

RUNPOPAN

Installation and User's Manual for Running
POPAN-5 on IBM PC microcomputers
under Windows 3.1/32S, 95, and NT

A. N. Arnason, L. Shar, D. Neilson and G. Boyer

Department of Computer Science
University of Manitoba
Winnipeg, Manitoba, R3T 2N2 Canada

Scientific Report,
Department of Computer Science

© 1998, A. N. Arnason

Release 2.2

October 1998

Table of Contents

Section	Page
INTRODUCTION.....	1
1.1 WHAT WAS ADDED IN POPAN-4 (AND STILL WORKS IN POPAN-5)	2
1.2 WHAT IS NEW IN POPAN-5.....	3
1.2.1 Maintenance Release 5B.2 (August 1998).....	4
1.3 CHANGES NEEDED TO RUN POPAN-4 POP FILES UNDER POPAN-5.....	4
1.4 POPAN DOCUMENTS	5
1.5 ACKNOWLEDGEMENTS AND PERMISSIONS	6
2. INSTALLATION, START-UP AND TESTING	7
2.1 COMPUTER SYSTEM REQUIREMENTS.....	7
2.2 INSTALLATION FILES.....	7
2.3 INSTALLING POPAN-5.....	8
2.4 GROUPS, SHORTCUTS AND DE-INSTALLING POPAN-5	9
2.5 TYPES OF POPAN FILES AND THEIR MANAGEMENT	9
3. AN INTRODUCTORY TOUR OF POPAN-5 FROM WINDOWS	11
3.1. RUN POPAN IN BATCH MODE:	11
3.1.1 The POPAN standard output (run progress) window	11
3.1.2 Browse the .LOG file from a run.	12
3.1.3 Line spacing and printing of output files.	12
3.1.4 Browse the .RES file from a run.	13
3.2 DESCRIPTION OF THE TEST SUITE.	13
4 USING THE RUNPOPAN INTERFACE.....	16
4.1 THE RUNPOPAN INTERFACE	16
4.1.1 Main menu choices	16
4.2 RUNNING AN EXISTING POP FILE FROM THE INTERFACE.....	18
4.2.1 Edit Window menu choices	18
4.2.2 Help Window menu choices	19
4.2.3 Results Window menu choices	20
4.3 BUILDING A COMMAND FILE FROM SCRATCH	22
4.3.1 Syntax of command statement statements and file specification.....	22
4.3.2 An example with use of templates.	23
4.3.3 Description of Template menus and operations.....	27
5. OTHER PROGRAMS AND WIZARDS CALLABLE FROM RUNPOPAN	29
5.1 THE ENTRY PROGRAM.	29
5.1.1 Background on POPAN history formats.....	29
5.1.2 An Introductory tour of ENTRY	30
5.2 THE DOS STAND-ALONE PROGRAMS.	34
5.2.1 Program JOLLY.....	34
5.2.2 Program CAPTURE.....	35
5.2.3 The MR Interface for running programs CAPTURE and JOLLY	37
6. ODDS AND ENDING.....	38
6.1 NOTES ON USE.	38
6.2 ERRORS AND WHAT TO DO ABOUT THEM.....	38
REFERENCES	39

INTRODUCTION

This manual describes how to install and use the IBM-PC version of the POPAN system. POPAN means POPulation ANalysis and is a system for the management and analysis of mark-recapture data, sometimes called capture-recapture or capture-mark-recapture data. The system includes capabilities for entering data as individual or grouped capture histories along with attribute information, for selecting subsets and groupings of the animals and sample times, and for obtaining parameter estimates (abundance, survival and recruitment rates) and goodness-of-fit tests using a very wide range of models for the sampling and population dynamics. The system also includes a very general statistics gathering capability and the ability to simulate sampling experiments for studies of precision and robustness.

POPAN was originally developed for IBM mainframe machines (POPAN-2) and was later updated to include many new analysis and testing options (POPAN-3). POPAN-4, released in November, 1995, added the ability to fit and test models with user-defined temporal constraints and covariate models based on a new unified theory (Schwarz and Arnason, 1996) for open mark-recapture models. POPAN-5 extends these capabilities to group constraints (e.g. between sex or age cohorts). Versions of POPAN-3 for VAX (VMS) and SUN (SunOS) machines were also developed. POPAN-4 and 5 were developed on the SUN (Sun OS, Solaris) and ported to the IBM/PC using Watcom FORTRAN compilers that permit generation of versions for several platforms (so far, IBM+Windows and IBM+OS/2). Starting with POPAN-3, we developed an interface called POPAN-PC for running POPAN on IBM-PC machines under DOS (release 1, 1988) or DOS or Windows (release 2, 1992). With POPAN-4, POPAN-PC was replaced by a new interface (called RUNPOPAN). The interface was redeveloped to make better use of windowing environments and to extend it across platforms (IBM+Windows. SUN+Xwindows). This IBM version of RUNPOPAN runs POPAN-5 on IBM-PCs under Windows 3.1 with the 32 bit extension (Win32S, supplied with the installation package) or Windows 95 and Windows NT (Version 4.0). The Installer (WINstall, from Dovcom Inc., Provo, Utah) works with all 3 systems.

POPAN-2 through POPAN-5 are batch systems. They are run in much the same way as large batch-oriented statistical analysis systems such as BMDP, SAS, or SPSS: that is, you can use a general editor (e.g. DOS/edit, Windows/Notepad) to type in a file of command statements and a file of the data to be analysed and then submit a job that causes the statistical system to be invoked, reading in your command file which in turn, causes the (raw) data to be read in and analysed according to the directions given in your command statements. The results of the analyses are written out to a file. After the job is finished, you can examine this output file and, if you like, direct it to the printer. POPAN, like SAS, actually writes 2 output files: a log file, that reflects back your command statements and any warnings or error messages, and a results file. Also like SAS, POPAN may also, if so directed by appropriate command statements, generate other files such as data files obtained from the original file by sub-setting and transformation operations. These files are written in a (binary) format that is only comprehensible to the statistical analysis system, but they can be used in place of the (raw) input data for subsequent jobs. The current implementation (RUNPOPAN) has an interactive user-interface, but many of these batch features are still preserved, at least in the background, so it is important to understand how the system uses the various file types: command files, raw and binary data files, and the log and result print files.

RUNPOPAN is a full-screen interactive user-interface that lets you carry out all the actions needed to run any POPAN job. You can create a command file, save it, run it, retrieve it, re-edit it, re-run it. After a run, you can retrieve and browse the log and results file for that run. The files are given the extension .LOG and .RES by the system. A command file has the extension .POP and is made up of paragraphs (whose names we denote in capital letters) that give you the ability to: CREATE a binary file from raw data, to SELECT a binary file and possibly a subset of its histories and sample times, to carry out a pre-programmed ANALYSIS, log-linear model TESTs, or collect STATISTICS using the most recently SELECTed file, and to LIST a file. A very general SIMULATE capability lets you generate and analyse data from replicated sampling experiments on a population you define. The new constrained model features of POPAN-4 and POPAN-5 are in a paragraph called UFIT.

Each paragraph starts with a paragraph name followed by a colon (e.g. CREATE:) and is followed by a number of statements of the form:

KEYWORD=KEYWORD_VALUES;

The paragraph is terminated by the null statement ;/. The RUNPOPAN interface lets you open an editing window for constructing a command (.POP) file made up of one or more such paragraphs. To add a paragraph you want to carry out; you select and fill in a template that lists each of the KEYWORD= statements and you can then specify the KEYWORD_VALUE. When a paragraph template is complete you can paste it into the command file currently being edited. You can also create or change paragraphs in the edit window directly by the usual edit operations (typing, copy, paste, etc.). When you are done, you can save the command file. The interface lets you run any of the command files you have saved, and, once the run is complete, it lets you load and scroll through a split window that contains the log (.LOG) file in the upper pane and the results (.RES) file in the lower pane. The interface lets you have several editing, template, and split windows open at once.

The RUNPOPAN interactive interface also gives you support in entering the raw data and accessing other Mark-Recapture programs. The ENTRY feature lets you enter raw data in one of several formats, and will construct the CREATE paragraph needed to convert the raw data to internal (binary) format. For some formats, it will also make a data file that can be used, without modification, as input to the JOLLY and CAPTURE programs. These 2 (DOS) programs can now be run from RUNPOPAN through a Windows interface called MRint. The ENTRY program and the MR interface are explained in Section 5 of this manual. In this release, ENTRY and MRint replace rather clumsy DOS programs that had been carried over from POPAN-3.

POPAN-4 will continue to be available from the POPAN web site for those without the resources (or need) to run POPAN-5 (see section 2). POPAN-5 contains all the capabilities of POPAN-4 as a subset of its capabilities (with some exceptions described below). We describe the POPAN-4 additions (1.1), and then the new features added in POPAN-5 (1.2) and, in section 1.3, the fixes needed to run POPAN-4 POP files under POPAN-5. Users of POPAN-4 who plan to rerun POP files in POPAN-5 should note the differences carefully. Then (1.4), we describe the manuals and their use and how to proceed with installing POPAN and learning to use it.

1.1 What was added in POPAN-4 (and still works in POPAN-5)

1. Backward compatibility: POPAN-4 binary files were not compatible with POPAN-3 and earlier binary files but all .POP and .RAW files were compatible (so users had to re-CREATE their binaries from RAW data; all other tasks ran as before).
2. Limits to the number of sample times that could be accommodated by various analysis methods were increased and reported in the .LOG file.
3. File specification in all paragraphs was made explicit by giving a relative path INPUT="TEST\VOLES.BIN"); or absolute path (e.g. INPUT='C:\POPAN4\TEST\VOLES.BIN');. Unit numbers are still accepted as before but require a shell script to associate units with files.
4. CREATE: raw data can be provided in-line following the paragraph (terminated by an \$ENDDATA record) rather than in an external (.RAW) file; raw data can be read using unformatted stream input (FORMAT=FREE) of blank separated fields; alternate forms of capture history data, including the CMR fixed length format, were allowed.
5. After a CREATE, a SELECT of the newly CREATED data is assumed so it is not necessary to specify SELECT (unless you want a subset or grouping of the data). SELECT keywords OMIT and GROUP allow specifying ranges of sample times (e.g. 5:10 instead of 5,6,7,8,9,10)
6. The LIST paragraph was extended to convert individual histories to grouped and to sort on cell number (a unique representation of the capture history). Cell number had been developed in POPAN-3 for the log-linear models (see Chapter 8 of AS&B) but is useful in general for checking data and doing goodness-of-fit tests. Default values for FORMAT and NFORMAT were allowed.

7. ANALYSIS and SIMULATE tables were expanded to two sets of tables to accommodate larger numbers of estimates. Keyword NFORMAT was added to custom format the second table.
8. The UFIT paragraph was added to allow user-specified constraints on the parameters. In POPAN-4, only constraints involving sample times are allowed; (in POPAN-5 constraints can also be across groups). Equality constraints can specify that parameters are constant across specified times; value constraints (e.g. birth rate=0) can specify selective closure conditions (sample intervals when births and/or mortalities are known not to occur). UFIT also allows the specification of temporal/group covariates and their use to model the capture, survival and birth rates. For example, you could enter the sampling effort for each sample and develop a model for capture rate as a function of effort.
9. SIMULATE was integrated with TEST and UFIT. You use a previous TEST (SAVE=TEST) or UFIT (SAVE=UFIT) as the definition of the analysis to apply to the simulated data (ANALYSIS=TEST replaced TEST=number and ANALYSIS=UFIT was added; pre-programmed ANALYSIS is still specified by ANALYSIS=number). SIMULATE now allows up 999 replications instead of 99. Some POPAN-3 keywords (mostly associated with TEST=number) no longer work.
10. SIMULATE can generate and analyse different population subgroups using the new keyword NGROUPS=g and the new GDEPENDENT keyword value for keywords of the form xMECHANISM. Independent analyses can be done on each group, giving a statistics and estimate table that has g*K rows for K-sample experiments. In POPAN-5, analyses with constraints across groups are possible through ANALYSIS=UFIT.

A fuller description of POPAN-4 was published in Arnason & Schwarz (1996). The complete references are given at the end of this manual.

1.2 What is new in POPAN-5

1. Backward compatibility: POPAN-5 binary files are fully compatible with POPAN-4 (but not earlier) binary files; you do not need to reCREATE them. Most POP files will run unchanged, but some minor changes are needed in UFIT and SIMULATE tasks (see next section for a list of these). There are no changes to the operation of the RUNPOPAN interface, except to fix a bug so it now works in NT. The template and help files have been updated to reflect the changes below.
2. A LENGTH keyword is added to all paragraphs to permit longer line length in the task command (POP) files (POPAN-4 ignored all characters beyond column 72 in the input POP files).
3. Extensive changes to UFIT permit the extension of constrained likelihood fits to multiple groups. Up to 9 groups can be defined based on attributes or histories (permitting year-class cohorts as groups). Linear or covariate constraints can be defined within or among groups (using a group-prefix extension of the POPAN-4 notation). Covariates may be time-dependent (the same for all groups) or time- and group-dependent.
4. Global constraints in UFIT allow quick specification of constant and closure constraints at all times (or a subset of times defined by new range restriction keywords) within and among groups. Additive constraints among groups can be specified on the natural, logit and log scale.
5. UFIT ADJUSTments for unequal sample time effects on survival and net or gross birth rates have been fully implemented in all constraint types (partially implemented in POPAN-4...now extended to multiple groups, including possibly different sample spacings in each group).
6. UFIT can be applied to history data read from a preceding SELECTed binary file (as in POPAN-4), but now it can also be applied to a table of sufficient statistics read in by UFIT directly. This permits model fitting when only summary data is available, but also permits powerful Mean Values analysis for investigating precision, bias and model selection properties (see below) as an alternative to SIMULATE.
7. UFIT with groups is fully integrated with SIMULATE with groups (see GDEPENDENT in previous section), so that group constraints can be applied to simulated data.
8. SIMULATE has added ATYPE=MARKING and ATYPE=HANDLING to compute age as the time since

an animal was first marked, or time since an animal was last captured. This permits investigation of marking and handling effects on subsequent capture or survival rates (what the fisheries literature calls Type-1 effects).

9. The preliminary report of SIMULATE is extended to include expected values for the sufficient statistics required by UFIT (and the J-D models in ANALYSIS). These can be SAVED to a text file for INPUT to UFIT permitting Mean Values analysis (the estimates, SEs, and model selection criteria are the expected values given the SIMULATE model mechanisms). SIMULATE can save the actual statistics from a single replication in a form suitable for INPUT to UFIT, and ANALYSIS can do the same using statistics formed for a J-D analysis of data from a SELECTed binary file.
10. Over 10 new examples have been added to the TEST suite to illustrate these new properties. These include 3 examples of Goodness-of-Fit testing from the TESTGOF suite (see TESTGOF.ZIP on the web site).
11. Extensive additions to the manual (AS&B, Ch 7) explain in detail how to use UFIT effectively and interpret its output.

A description of POPAN-5 is published in Arnason & Schwarz (1998) The final draft of this article, which emphasises applications of POPAN to banding and age-structured data, is available on the POPAN web site.

1.2.1 Maintenance Release 5B.2 (August 1998)

This release makes some fixes to the POPAN program and includes enhancements to the RUNPOPAN interface

1. Errors in reading histories in CREATE were corrected (the error occurred when using FORMAT=FREE with injected animals) and the code generally revised to make it more robust. See (revised) online Help (Raw histories) or AS&B Sec 2.1 for details of correct history formats.
2. This release fixes errors in UFIT that occurred if no SPCON (survival constraint) was specified and if sample sizes of 0 occurred.
3. The RUNPOPAN interface has been enhanced with a more flexible results split window (each pane is re-sizeable) and the DOS-based ENTRY program has been replaced by an interactive G(see Section 5.1 below)
4. Program JOLLY (Pollock et al. 1990) for open population analysis and program CAPTURE for closed population analysis (Rexstad & Burnham, 1992) can be invoked from RUNPOPAN through a Windows interface (McGlinchy and Barker, 1998) . Data for these programs can be generated by the Entry wizard.

1.3 Changes needed to run POPAN-4 POP files under POPAN-5

1. SIMULATE tasks: SAVE=YES is no longer valid for saving histories from the simulation. You must change this to SAVE=HISTORIES;
2. UFIT tasks:
 - a) SVALUES has been changed to IVALUES (for specifying initial parameter values) but otherwise the syntax is unchanged.
 - b) TRACE, if specified, must be an 18-bit string (not a 16-bit string) and the meaning of the trace bits has changed. Use TRACE=100000000000000000; or consult the manual.
 - c) Add the keyword AUTO=NO; so that the automatic constraints on capture rate at the first and last time are NOT imposed: this was the default in POPAN-4 but AUTO=YES is the default in POPAN-5.
 - d) Change ANALYSIS=1 to ANALYSIS=11 (all initial ANALYSIS choices must be of the Jolly-Dickson full-model type: see Help: Analysis Numbers)

1.4 POPAN Documents

There are two manuals needed for using the POPAN system. This Installation and User's (I&U) Manual describes how to install the system and how to use it with or without the interactive interface. It is available as a MS Word (version 6 or later) document or as a Postscript (.ps) or Adobe Acrobat (.pdf) file. A compressed file with these versions can be downloaded via the Internet or it can be ordered on diskette. The Postscript version can be viewed if you have a PS viewer (such as Ghostview) or it can be sent directly or via the viewer to a Postscript printer. The Acrobat version can be viewed by most Internet browsers with the Adobe Acrobat Reader plug-in. For example, in Internet Explorer, you can use File...Open and Browse for files of type .pdf on My Computer. The .pdf file should provide more reliable printing than the .ps format. Paper copies of the manuals are available for a charge; you can order them using the ORDERFRM.TXT from the POPAN web site or request an order form from the authors. The I&U manual is not a stand-alone reference on the capabilities of POPAN nor does it attempt to describe the interpretation of POPAN output; it is just intended to get you started. To understand how to input and manage data and its (descriptive) meta-data, and how to use the other paragraphs effectively to analyse the data, you need to read the POPAN manual. The manual (Arnason, Schwarz and Boyer, 1998 referred to as AS&B) The manual has been thoroughly revised for release with POPAN-5; extensive new material has been added on UFIT (Chapter 7) and the description of SIMULATE (section 6.1) has been radically re-arranged. Other chapters required only minor revisions over the version released with POPAN-4. The most important chapters for a first reading are the general Introduction to POPAN (Chapter 1), CREATE (in Chapter 2), SELECT (in Chapter 3), ANALYSIS (Chapter 5), and UFIT (Chapter 7). The complete manual is over 300 pages long.

Users can, however, get started without reading AS&B, by following this Installation and User's manual. We explain how to obtain POPAN-5 materials and install the system in section 2 of this manual. Most of the material in section 2 has been extracted into the README.1ST and INSTALL.TXT documents available as text documents on the POPAN web site, so you may already have done the reading. Once you have done the install, you can follow Section 3 of this manual to carry out the initial test of POPAN in batch mode using the supplied command (.POP) files in the Test Suite that comes with the installation package. Section 4 guides you through an introductory tour of the RUNPOPAN interface, showing you how to compose a .POP file that will read some data, CREATE and save a .BIN binary file for the data, and run some analyses on it. With this basic introduction, you should be able to carry on doing your own analyses. The example POP files in the TEST directory provide examples that you can adapt and there is on-line Help available in RUNPOPAN that explains the operation of each paragraph including details of keyword definitions, and examples.

Once you begin to use more subtle or complex features of POPAN (especially SIMULATE, TEST and UFIT) you should consult the full manual for a fuller discussion of how to use these capabilities effectively and how to interpret their output. The manual is organised by paragraphs and can be consulted as you need it, rather than all at once. Each Chapter deals with one (sometimes two) paragraphs. The first section of each chapter deals with the technical aspects of using the paragraph while later sections deal with the methodologies used and interpretation of output. The organisation of Chapter 5 (The Analysis Paragraph) is typical:

Section	Title
5.	Population Analysis
5.1	Programmed ANALYSIS options
5.1.1	Introduction to ANALYSIS
5.1.2	Description of the ANALYSIS Paragraph
5.1.3	ANALYSIS Task Definition
5.1.4	Examples of ANALYSIS
5.2	Interpreting the Estimates from Models without POOLING
etc....	

At a first reading, the user should concentrate on these first sections. The Introduction (e.g. 5.1.1)

explains the scope and purpose of the paragraph, the Description (e.g. 5.1.2) describes the background necessary for understanding how the paragraph works, and the Task Definition (e.g. 5.1.3) gives complete descriptions of how to specify each Keyword for the paragraph. The Examples (e.g. 5.1.4) are heavily commented to explain what they do and are taken from the TEST Suite. The essential contents of the Task Definition and Examples, along with a brief statement of the scope and purpose are also available as on-line HELP (from the Help menu) in RUNPOPAN. The most important chapters for a first reading are the general Introduction to POPAN (Chapter 1), CREATE (in Chapter 2), SELECT (in Chapter 3), ANALYSIS (Chapter 5), and UFIT (Chapter 7.1).

1.5 Acknowledgements and permissions

This version of RUNPOPAN was implemented using the freeware Wx graphical user interface (Smart, 1995) and Borland C++. The POPAN program is written in FORTRAN and compiled using WATCOM FORTRAN 77. Documents were prepared in Microsoft Word Version 6.0 for Windows. These product names are trademarks or registered trademarks of their respective companies and are hereby acknowledged.

The programs JOLLY (Pollock et al. 1990) CAPTURE (Rexstad et al. 1991) and MRInt (McGlinchy et al., 1998) are freeware, distributed with permission of their authors. As with POPAN, you should not copy and re-distribute this software, but rather, direct others to the respective web-sites for obtaining current versions (see References for web addresses). You may not charge users for copies of these programs or their distribution materials other than a reasonable fee for copying and delivering (e.g. cost of disk and postage). You may not alter the underlying code, or otherwise modify these programs without written permission of the respective authors.

The authors gratefully acknowledge the long-term support of the Natural Sciences and Engineering Research Council of Canada who supported this work through operating grants and many summer NSERC studentships. Without this support, the development of POPAN and its free distribution would have been impossible. The preparation of POPAN-5 for distribution and the further development of POPAN is supported by a grant from Manitoba Hydro.

2. INSTALLATION, START-UP AND TESTING

2.1 Computer system requirements

POPAN-4 ran on IBM 386 and later machines and came in co-processor and non-coprocessor versions. It ran under Windows 3.1 in 386 enhanced mode (but required installation of the Win32S upgrade). It would run quite comfortably on a machine with 8Mb of memory and required only about 3 Mb of hard disk space.

POPAN-5 is only supported on co-processor machines (486DX and up) running full 32-bit operating systems (Windows 3.1+Win32S, Windows 95, Windows NT). POPAN-5 does NOT (yet) support long file names, so all files referenced and used by POPAN must conform to the old DOS-FAT convention of 8-letter file names with 3-letter extensions (e.g. TESTANAL.POP). POPAN requires about 8 Mbytes of disk space to install, and will take up about 4.25 Mb after the install (not including manuals). It needs about 2Mb of memory to load and then may allocate up to 16 Mb of additional (stack) memory dynamically as it runs. Thus you may need more hard disk space to run if you are using disk-based virtual memory. We have experienced no memory problems in running on machines with 16Mb of RAM and 20 Mb of free hard-disk space.

2.2 Installation files

The POPAN5 distribution package is available on diskette or may be downloaded via the Internet from the POPAN web site (<http://www.cs.umanitoba.ca/~popan>) or the POPAN anonymous FTP site (<ftp://ftp.cs.umanitoba.ca/pub/popan>). If using an FTP program rather than a web browser, connect to [ftp.cs.umanitoba.ca](ftp://ftp.cs.umanitoba.ca) and logon (using ftp as the userid and your full email address preceded by a minus sign as the password: e.g. -ename@eaddress). After getting the ftp prompt, change directory to pub/popan/popan5 or pub/popan/utilities. All files MUST be transferred in binary mode, except the README.1ST and other text (.TXT) files. The distribution involves a number of files, most in compressed form, which should be copied to a \TEMP directory on your hard disk in preparation for installation. The README.1ST and INSTALL.TXT are available in uncompressed form and simply repeat the essential introductory and installation material from this manual for those who are downloading and installing before obtaining this manual. You should download the following:

(a) In the /utilities directory

(you don't need this if you have WinZip or an equivalent .ZIP file extractor):

PKUNZIP.EXE The decompression program needed to unload the .ZIP files

(b) In the /popan5 directory:

POPAN5.ZIP The POPAN5 executable and table file. Also includes the JOLLY program (01/24/1991 ver) of Pollock et al. 1990 and the CAPTURE program (07/13/1991 ver) of Rexstad and Burnham Approx. 1.1 Mb

POP5RUN.ZIP The RUNPOPAN interface and installation program including the template (.TAB) files and help (.HLP) files and the TEST Suite (.POP) files; also includes the 3 text files (README.1ST, ORDERFRM.TXT and INSTALL.TXT along with the MRinterface program of McGlinchy and Barker (03/01/1998 ver 1.1) and the Entry Wizard. Approx. 1.1 Mb

POP5IUM.ZIP The Installation and User's Manual as a Word 6.0 (.doc) document, and as a PostScript (.ps) document, and an Adobe Acrobat (.pdf) read-only file. Approx. 750 Kb

POP5MAN.ZIP The POPAN manuals (AS&B) in individual files for each chapter plus a Master document. These are MS Word 6.0 for Windows files. Includes a readme (text) file that explains how to access the documentation provided. Printed copies of the manuals are also available (see order form ORDERFRM.TXT). Approx. 500 Kb

POPAN should be set up under its own directory C:\POPAN5 on the C: hard drive. It can be set up under another directory or drive if necessary. The install program in POPANRUN.ZIP automates this and creates a Windows group (folder) and icon for launching the RUNPOPAN interface. In Windows 95 or NT, POPAN-5 is added to the Start... Programs... menu. If you have a previous version of POPAN on your machine, it is probably under a directory called C:\POPAN4 or similar. It is not necessary to delete this version before installing POPAN5 and the installation will not affect any of the existing system or user files. You may wish to keep old user files around for re-analysis with POPAN5 but the system files can (eventually) be deleted. NOTE: the current version of the installer does not work correctly unless you let it create the folder. Thus if you install to C:\POPAN5, this folder should NOT exist prior to the install. If you have a folder with this name, either rename the folder or delete it before installing.

2.3 Installing POPAN-5

The installation is done from Windows using the Run dialog box (under the Start button). We assume you are familiar with basic Windows operations (create / delete / rename a folder; delete files; Run an application). We also assume that you already have access to this manual in printed or electronic form (step 3. in INSTALL.TXT explains how to do this). The steps in the installation are:

1. Make sure your computer meets the minimum standards (section 2.1) for installation of POPAN and that the directory you want to install to (e.g. C:\POPAN5) does NOT currently exist.

2. Create an empty folder on your hard drive called \TEMP (or empty it if it already exists). Place in this directory the ZIP files you need, by copying from the distribution diskettes or by downloading via the Internet. You must have AT LEAST the first two ZIP files listed above or the installation will fail. Note that installation MUST be carried out from a folder other than the final POPAN folder so DON'T download into a folder with the name POPAN5. This final folder should NOT exist prior installation, so DON'T create the folder yourself: let the installation program do it for you.

3. Run the PSETUP.EXE program. In Windows 3.1, this is done from the Program Manager by choosing ...Run from the File menu and typing: C:\TEMP\PSETUP.EXE in the Command Line box. In Windows 95 or NT, Click the Start menu at the lower left of your desktop, and choose Run. Type the same command in the Open box. Click the OK button to start the install.

A window will pop up and ask for the directory into which POPAN5 is to be installed (the default is C:\POPAN5). It then proceeds to transfer the files. If you get a STOP message with the request "insert disk #1" it means that a file it is trying to transfer was not in the C:\TEMP directory. Check that you unzipped all the required files in step 2.

4. After the installation is complete, you will get another chance to read the README.1ST file (but you have the option to skip this). When you are returned to the Windows Desktop, you will have a POPAN5 folder open on the desktop. (Windows 3.1 users will have a new Group (folder) called POPAN5 in the Program Manager Window). If you open the POPAN5 group folder, you should see that it contains icons for the RUNPOPAN (interface) and ENTRY (data entry) programs and also for JOLLY (simple open-population analyses), CAPTURE (closed-population analyses) and the MRinterface to these 2 (DOS) programs, but don't worry if any of them are missing (see 2.4 below).

5. Follow the introductory tours of POPAN in section 3 and 4 of this manual. If any of the programs won't launch or if you get missing files messages, consult the File Checklist in the INSTALL.TXT document (which should now be in both your TEMP folder and your POPAN5 folder), to ensure that all files were installed correctly.

6. Copy the POP5MAN.ZIP and POP5IUM.ZIP files from the TEMP folder to a convenient permanent folder (possibly a Manuals subfolder of your POPAN5 folder). You can decompress them now or later when you need to consult them again.

7. Delete the contents of the TEMP folder.

2.4 Groups, Shortcuts and de-installing POPAN-5

Windows 3.1 users who don't want a separate Windows Group (folder) for POPAN5 can open the POPAN5 Group created by the installation program and drag the POPAN icon to another Group. Then close the POPAN5 Group but keep it selected and choose File...Delete from the Program Manager menu. In Windows 95 or NT, you have the opposite problem (the POPAN5 folder placed on your desktop by the installation will not re-appear next time you log on); if you want a Shortcut on your desktop, use My Computer to locate the POPAN5 folder and right-click on RUNPOPAN.EXE (**not** on POPAN.EXE) and choose Create Shortcut. You can then drag the shortcut to the desktop or any other folder. Double clicking the Shortcut launches the interface; without a Shortcut you can use Start...Programs...Popan-5...RUNPOPAN (which has been added to the Programs list by the installation program). You **must not** attempt to launch POPAN.EXE by shortcut or by double clicking on its icon as it always requires a POP file as a command line argument (which cannot be supplied by launching in this way; if you do launch in this way, a Watcom window will open and a lot of output will flash past; when it stops, just close the window). In section 3 we show you how to launch POPAN directly using Run so that you can Run an existing POP file as a batch job without using the interface.

The MRint and ENTRY programs can be launched directly (by clicking on their icons in your POPAN folder or from your POPAN-5 group (Win 3.1) or from Start...Programs...Popan-5 (Win 95 and NT

If you want to uninstall POPAN you just delete all the files in your POPAN5 folder (e.g. C:\POPAN5) and its sub-directories. If you delete this folder, you should also delete your POPAN5 group (Win 3.1) or the Popan-5 group in Start...Programs menu (Win 95/NT: use Start...Settings...Taskbar...Start Menu Programs...Remove). POPAN modifies no system files and does not change system .DLLs but it does add a small **MRint.ini** file to the Windows (or Winnt) directory for remembering your Path information. You can delete this if you wish (the MRint program will recreate it automatically next time it is run) .

2.5 Types of POPAN files and their management

POPAN keeps a large number of files. Some are system files needed for running POPAN and some are files created by you as you enter data and carry out analyses on them. POPAN uses a strict file extension convention and it is important that you adhere to this so that the interface can recognise the file type. The types are

(a) System files.

These reside in your main POPAN5 directory (typically, C:\POPAN5) and should not be deleted, edited, moved or renamed (we also recommend you don't store user-created files in this directory).

- | | |
|-------|---|
| *.EXE | The are the executables for the POPAN program (POPAN.EXE) and interface (RUNPOPAN.EXE). |
| *.TAB | These are table files need by POPAN (POPAN.TAB) and the interface. The remaining .TAB files define the menu and status line text (MENU.TAB) and the template layouts. |
| *.HLP | Text files for on-line Help from the RUNPOPAN interface. |

(b) User-created files.

These are typically kept in a subdirectory of your main directory, such as C:\POPAN5\MYSTUFF. The first two file types are created directly by you (by editing a text file) while the others are created by the POPAN program. All, except .BIN and .OUT files, are plain text files.

- | | |
|-------|---|
| *.POP | POPAN command files. These define the POPAN tasks you want carried out using a specific command syntax made up of paragraphs and statements. |
| *.RAW | The raw data (not including meta-data) needed by the CREATE paragraph. They are also used by ADD to specify additions to an existing binary file. |
| *.BIN | The binary output files generated by CREATE, but also by SELECT: with SAVE= and by |

	ADD. These files combine meta-data supplied in keywords with the raw data. They are not readable from most text editors.
*.LOG	The unit 6 (parse-time) output from running a .POP file that reflects back the command statements along with any error messages from the command parser.
*.RES	The unit 8 (execution time) output from running a .POP file. LOG and RES files are given the same prefix as the POP file that generated them (e.g. SKUNKS95.RES if generated by SKUNKS95.POP).
*.OUT	Files of saved table output resulting from using the keyword DATASET= in the LIST, ANALYSIS, or STATISTICS paragraphs. These are saved in binary form (for the moment...plans are to change this in a later release of POPAN) and are not (human) readable or text editable; they are intended to be read by a user-written program as explained in the POPAN-2 manual.
.TBL	A text file of saved statistics in a form suitable for INPUT by UFIT. These can be generated by a SAVE in ANALYSIS or SIMULATE or the user can compose a suitable file from any text editor.
.ESV	Entry Save files are created by entry so that you can re-start the wizard and resume entering data from the point you last saved
.REL	Raw data files in Release format (Burnham et al. 1990); also compatible with Mark (White, 1998) encounter format LLLL. For input to programs JOLLY and CAPTURE.

When you begin to use POPAN on your own data, we suggest you create further subdirectories of your POPAN5 directory for each 'project'. An example of a 'project' is the \POPAN5\TEST files which contains the Test Suite of files including raw data (.RAW) files, the POPAN command (.POP) file for converting it to binary form and all the command files for further analyses of the data and any binary files derived from the original binary file (by subset selection and sample time grouping). As we shall see, a 'project' subdirectory might also contain any of the print files (extension .LOG and .RES) resulting from running the command files. We suggest that you create a new subdirectory for each dataset that is analysed separately from other data (e.g. POPAN5\SKUNKS, POPAN5\WALLEYE, etc.). We especially recommend that you AVOID placing user (.RAW, .BIN, .POP, .LOG, and .RES) files in the \POPAN5 directory itself: this should contain ONLY the necessary executables (.EXE) and support (.TAB, .HLP) files.

You will want to delete files from your 'project' subdirectory from time to time. Think twice before deleting .POP or .RAW files, and .BIN files can be expensive to regenerate. But if you have no immediate further need for output (.RES or .LOG) files or have just printed them, delete them. They are easily re-generated if needed and can take up a lot of disk space if left to accumulate. There is no capability to delete files from within RUNPOPAN so this must be done from Windows or from the DOS prompt.

3. AN INTRODUCTORY TOUR OF POPAN-5 FROM WINDOWS

This tour will demonstrate the use POPAN in batch mode (without using the RUNPOPAN interface). We will introduce use of the interface in section 4. In both sections, the figures and instructions are for Windows 95 and are essentially the same for Windows NT (version 4.x). Users of earlier versions (Windows 3.1 or NT 3.x) should be able to follow this, although the details of window and file management are slightly different. If you have difficulties, you can obtain the Installation and User's manual for POPAN-4 and follow the identical tour presented there using Windows 3.1.

3.1. Run POPAN in batch mode:

To use POPAN as a batch program from Windows without using the RUNPOPAN interface, you must have an existing command (.POP) file which should be in a subdirectory of your POPAN5 directory, and you will need to use your own editor program to view the result (.LOG and .RES) files from the run. We will use one of the Test Suite programs and the Windows Notepad accessory for our first run. This provides a check that the POPAN installation in step 5 worked, independent of the interface.

From the Windows Program Manager select **Run...** (under the Program Manager's **File** menu in Win3.1; on the **Start** menu in Win 95 and NT) and, from the Run dialog box, type the following Command Line:

`c:\popan5\popan test\testcr8v`

(Note that the general form of this is: `popanpath\popan subdir\popfile`

where *popanpath* is the full drive and path where the POPAN.EXE resides, and *subdir* is a subdirectory of this path where the .POP file popfile.POP resides. The extension .POP does not need to be given. If the .POP file is not in a subdirectory, *subdir\popfile* can be replaced by an absolute path (e.g.

`c:\skunks\july\firstrun` where firstrun.POP is the desired command file).

3.1.1 The POPAN standard output (run progress) window

If POPAN is properly installed, POPAN should load (this can take 5 or 10 seconds on a slower 486 machine) and open a standard full-screen output window labelled WATCOM 386 Windows Application. It contains a (child) Standard IO (Input/Output) window, shown in Figure 3.1. POPAN writes a progress report to that window consisting of the names of the paragraphs it encounters. The TESTCR8V.POP command file does a CREATE using data supplied in-line and then does an ANALYSIS using the Jolly-Seber full model. The data is the voles data of Pollock et al. (1990). When the run has completed successfully, the Standard IO window should appear as shown in Fig. 3.1.

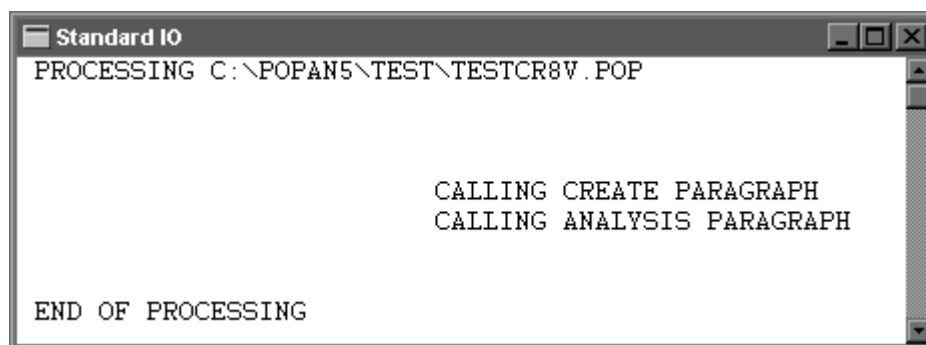


Figure 3.1. The standard IO window after running the TESTCR8V.POP program

Once the END OF PROCESSING message appears, the job is complete. At this point, close the window by clicking on the Close box (upper right-hand corner of the Watcom (parent) window (not shown in Fig 3.1) or select **Close** from the drop down menu (upper left corner of the window). **NOTE:** until you close this

window, any files generated by the POPAN run (in this case, the VOLES.BIN output binary file and TESTCR8V.LOG and TESTCR8V.RES, the log and results files) may appear to be missing or empty if accessed by another Windows application (e.g. File Manager, My Computer, or Notepad).

3.1.2 Browse the .LOG file from a run.

Now launch the Notepad accessory (it's in the Accessories folder on your Windows desktop). Use File...Open and navigate to the \POPAN5\TEST directory and open the TESTCR8V.LOG file. The first screen you see has a source identification consisting of date/time stamp and the name of the POP file used to generate the LOG file, followed by title/author information. Maximise the window using the control button on the Title Bar so that the Notepad window takes over your entire screen.

Scroll down past the title/author page. You will see a page of notes about the current release, ending with a disclaimer. These 2 pages, apart from the source identification, are always the same for POPAN .LOG files. The next section of the log file begins THIS VERSION OF POPAN-5 COMPILED WITH: and then gives a table of limits on the number of sample times etc. permitted by various POPAN tasks. It is important to note these limits for your version: other versions may have different limits. The next page begins with the title line: POPAN 5 --- ANALYSIS OF CAPTURE-RECAPTURE DATA . Following this you will see the command statements (reflected back, with statement numbers) and, if there are any errors that were detected by the parser, there will be warning or error messages advising you of the problem encountered. After you have scrolled through the command statements, you will find the END OF RUN STATISTICS (if the job did not fail catastrophically) starting with a SUMMARY OF DIAGNOSTICS GENERATED. This gives the total number of errors and warnings generated. Only those related to statement syntax appear in the .LOG file. An example is the statement you can see just above the end of run section that says:

POP340W -CREATE PERFORMED BUT NO SELECT -- DEFAULT SELECTION IS ASSUMED.

It begins with an error message number (POP340) and severity level (W for warning) and then a statement of the error or warning and action taken. If there are more messages counted in the SUMMARY than appear in the .LOG file, it means that there are additional (execution time) messages appearing in the .RES file.

Following the SUMMARY OF DIAGNOSTICS is a SUMMARY OF PARAGRAPHS ENCOUNTERED. In this case, the program should have encountered and executed 1 ANALYSIS, 1 CREATE, and 1 SELECT paragraph. The SELECT paragraph was generated by POPAN and does not actually appear in the .POP or .LOG file. Usually a quick look through the .LOG file, to check that there were no serious errors and that the expected paragraphs were executed, is all that is necessary.

3.1.3 Line spacing and printing of output files.

Note, as you scroll through the .LOG file, that there are no blank lines, and the first character of each line is a FORTRAN carriage control character. These codes determine the line spacing when printed on some systems: 1 = new page, blank = new line, + = no line (i.e. overstrike), 0 = double space, - = triple space. For example, in Unix, using the **fpr** filter program generates the appropriate line spacings. In a later release we will provide an fpr filter for use with Windows when printing. For the moment, it is up to you to use your editor program to print the .LOG file. Use **File...Print Setup** to select your printer and **File...Print** to print the file. Log files have lines no longer than 80 characters so should print without line wrapping (exceeding the printed page line width) provided margins (**File...Page Setup**) and default printer font are set appropriately. The printed copy will include the carriage control characters at the beginning of each line. This makes for compact printing but the pagination won't be very logical and the output won't be very pretty.

When you have finished browsing and/or printing the file go to Notepad's **File...Open** menu and open the corresponding result file TESTCR8V.RES. Note that .RES files also have carriage control characters but use up to 133 character lines. To print these files on ordinary paper (8.5 inch width) you may need to set your printer to compressed or landscape mode or choose a small font (e.g. courier 16.67 pitch). On some

printers, this can be done from the printer's control panel, while on others it has to be done using a Page Setup from the application of printer driver. The files should be printed using a fixed pitch font with margins chosen to permit 133 characters per line without line wrapping. Much of the .RES output is in table form which will be out of alignment and difficult to read if these conditions aren't met. Notepad (and the RUNPOPAN interface) provide horizontal scrolling so the long lines don't create a problem when browsing on the screen. Many RES files are too big to open with Notepad; if you plan to browse RES files frequently from outside the RUNPOPAN interface, you may wish to set up an association for the .RES extension with Wordpad (see Associations in Windows Help) or some other simple text editor.

3.1.4 Browse the .RES file from a run.

Scroll through the TESTCR8V.RES file loaded into Notepad. Make sure your horizontal scroll bar is set so you are seeing the leftmost part of each line. Each paragraph output begins with a paragraph box (enclosed in asterisks) and the title lines for the paragraph. There may be up to 4 title lines, identifying, in order, the header title from the binary file (specified in CREATE: NAME=), the SELECT title, the task title (e.g. ANALYSIS: TITLE=), and an options / model title (written by POPAN). Scroll through the CREATE output; there are only a few lines indicating that the binary file was successfully created from 40 grouped histories representing a total of 311 animals. There are also two warnings and some feedback on the sample time ranges encountered. This is followed by a SELECT which is performed by default if no SELECT follows the CREATE. The output is quite long in this case because none of the options to suppress parts of the SELECT output are used. It reflects back the complete header information, including the absolute sample times (week number). Note that the vole experiment was not done at equally spaced times. The absolute times were taken from the midpoints of the sample time intervals reported by Pollock et al. (Table 4.7, p.29). Continue scrolling until you come to the ANALYSIS paragraph. The output consists of a STATISTICS table (preceded by its definitions) followed by the ESTIMATE TABLE. Some of the definition tables can be suppressed by the use of appropriate command statements and you will want to use this feature once you are familiar with POPAN's output. Scroll through the output, referring to Section 5.1 of AS&B for a description of the output. This is a simple Jolly-Seber model analysis with birth and death (as indicated in the last title line). The statistics are the same as those given by Pollock et al. (Table 4.7, except for a correction to their R(1) and S(6) value) apart from changes in notation, and the estimates are the same as those given in their Table 4.8. See Pollock et al. for a discussion of the results, including goodness of fit tests (tests like those in their Table 4.9 are carried out in POPAN using ANALYSIS: Number=39; see TESTAN39.POP for this example).

Scroll or page down until you are positioned so the ESTIMATE TABLE is at the top of your screen. You should see the column headings AH(I), MH(I), NH(I), S(NH(I) | N), etc. The table is quite wide and you will have to use horizontal scrolling to see both halves of it. The estimates for abundance (NH(I)) are very good, having a CV (= S(NH) / NH) of under 5%. Those for survival (PHI(I)) are also very good though not quite as precise, and all are admissible (greater than 0 and less than 1.0) as are those for births. This quality of estimate is not typical for the Jolly-Seber model but here results from extremely high capture rates (N(I) / NH(I)), mostly in excess of 0.9 (90%). For more typical results try the suggestions for further test runs below. The Jolly-Seber model is discussed in Section 5.2.1 and 5.2.2 of AS&B.

3.2 Description of the Test Suite.

Follow the procedures above with any of the other task files in the Test Suite. When you do so, take a careful look at the commented Keyword statements in the log file, or use Notepad to preview the .POP file, to see what the task does. Although there is a logical grouping and ordering to these tasks, you can, if you choose, run these in any order because the binary files are also supplied with the TEST suite (as are the .RAW and .POP files for re-CREATEing them, if you wish). We recommend that you try at least the first two or three of these, in the order given below, before proceeding to the next section.

Here is a brief description of the Test Suite tasks currently available. The first set creates and uses the

voles data from Pollock et al. 1990.

TESTCR8V.POP	The CREATE program for the voles (which you just ran). CREATES the VOLES.BIN file needed by the next two tasks using in-line, group-count data.
TESTAN39.POP	Goodness-of-fit tests for the voles data that reproduces the results in Table 4.9 of Pollock et al. 1990 using ANALYSIS: Number=39.
TESTUFIT.POP	Fit some constrained-parameter models to the (ungrouped) voles data using UFIT

The next set of tasks create and analyse a data set with full meta-data (attribute groups and sample time definitions). They create and use the BURN24.BIN file:

TESTCRE8.POP	Create the BURN24.BIN binary from the BURN.RAW. These represent 24 sample occasions taken morning and evening, twice a week.
TESTANAL.POP	Jolly-Seber model ANALYSIS of the BURN24 data, including an example of customising the formatting of the Estimate Table.
TESTSTAT.POP	Some examples of STATISTICS collection for goodness-of-fit and assumption testing using the BURN24 data; also includes generation of the (doubly subscripted) recovery matrix statistics. Note that the .RES output from this is quite long and Notepad will probably refuse to open it. Try a different editor or list it from the DOS prompt using : type teststat.res more

The next set creates and uses a subset of the BURN24 data:

TESTSELS.POP	Demonstrates attribute selection and sample time grouping using SELECT to reduce the BURN24 data to 12 samples (stored in BURN12.BIN) of an age-species sub-set.
TESTANPO.POP	Apply some Jolly-Dickson (parameters constant in time) ANALYSIS models to the BURN12 data and also try pooled ANALYSIS (Reduced Capture History moving average smoothing).
TESTLIST.POP	Various LISTings of the BURN12 data.

The next set are 3 programs taken from the TESTGOF suite (see TESTGOF.ZIP on the POPAN web site: this workspace includes various POP programs and a stand-alone (DOS) program for carrying out a number of tests of assumptions. The first 2 are applied to the BURN12 data but are perfectly general and can be re-used on any data by changing the INPUT statement in the first SELECT. The tests are well discussed in the Appendix (by A. Viallefont) in Cooch et al. (1997)

TESTR3S.POP	Test (mainly) of equal survival using ANALYSIS: NU=39
TESTR2C.POP	Test (mainly) of equal capture rate using ANALYSIS: NU=39
TESTSIMP.POP	shows how to use SIMULATE to investigate the power of a test to detect a given level of assumption failure.

The next set uses simulated data to demonstrate more advanced POPAN features. These can take quite a bit longer to run (minutes instead of the few seconds for the tasks above) and they produce more output ; we suggest you defer trying these until later.

TESTSIMU.POP	A replicated SIMULATE experiment to explore the effect of various violations of assumptions on the bias and precision of Jolly-Seber estimates.
TESTTEST.POP	Generation of a single replication of a population and sample with known parameters using SIMULATE followed by the use of TEST to see if it can select the correct model. It also shows how to implement the Pollock-Robson test for births

(recruitment) using ANALYSIS: Number=39;

TESTUSIM.POP A replicated SIMULATE experiment with known model parameters and no assumption violations to explore the ability of UFIT to fit models involving covariate models for the parameters.

The next set of programs illustrates advanced uses of SIMULATE and UFIT. They are discussed in some detail in Section 7.2 of AS&B. We suggest you defer running these until you have read this material. The next 2 programs create and use the European Dipper data (taken from Cooch et al. 1997 but more fully discussed in Lebreton et al. 1992)

TESTEDCR.POP CREATes a 2-sex (M and F) binary file ED.BIN and analyses each sex separately
TESTEDUF.POP Gives several examples of grouped UFIT with covariate and global constraints.

The next 2 programs create and use the Female Age-structured data (taken from Cooch et al. 1997)

TESTFACR.POP CREATes a binary file of annual band data FAGE.BIN and analyse each year-class separately
TESTFAGE.POP Use UFIT with groups to illustrate age-class model fits.

The next 3 programs do Mean Values analysis by saving the expected statistics table from a SIMULATE task and using it as input to a UFIT task.

TESTMVAL.POP Mean Values example with ungrouped data to illustrate parameter confounding
TESTSIMG.POP Example of SIMULATE with GroupDEPENDENT mechanisms and SAVE of the grouped Statistics table (to GRSIM.TBL)
TESTUFIN.POP Statistics INPUT of the edited Statistics table (to GRSIMED.TBL) by UFIT

4 Using the RUNPOPAN Interface

Running POPAN with the RUNPOPAN interface provides additional support when you are preparing your own data and command files. The interface provides you with an editing window that lets you create new, or modify existing, command files. This editor also lets you pop up template windows that list all the keywords for a given paragraph and allows you to enter the keyword values. The template includes short help comments for each keyword, and more complete on-line help is available for each paragraph through a **Help** menu. When you have filled in a template, you can paste the contents into your editing window. You can then fill in another template of the same type, or open another template or return to the edit window to edit its contents or run the program. You proceed in this way, building a POPAN command (.POP) file of one or more paragraphs.

When you are finished preparing your .POP file, you can submit (run) the program. The same standard IO window will appear (Fig. 3.1) as if you had run it in batch as described in Section 3. When the program finishes and you have dismissed the window, you can load the log and results file into a split window that makes it easy to browse the output from your run. You can repeat this edit-run-browse cycle as often as you wish.

We will start by describing the interface and then give an introductory tour of the interface, using it to browse, modify, and run an existing .POP file without using the templates. Then we will summarise the main syntactic rules for the POPAN command language. You need to know these because you are free to edit the contents of the window in any way you want and the editor and templates do not guarantee syntactically correct paragraphs; they just provide an aid in constructing them. Finally, we will give a second tour of the interface, this time using the templates to construct a complete command file from scratch and run it.

4.1 The RUNPOPAN Interface

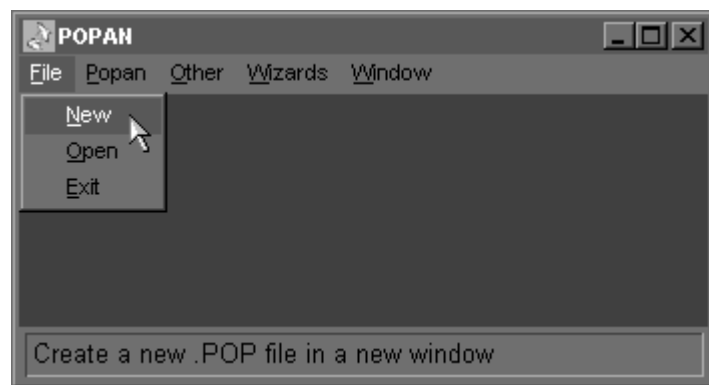


Figure 3.2: The RUNPOPAN Windows interface showing the main menu and the File drop-down menu.

To start the RUNPOPAN interface use Start... Programs... Popan-5...RUNPOPAN or simply open the POPAN-5 folder and double click on the Runpopan.exe icon. (Win 3.1 users can open the POPAN-5 Group and double click the RUNPOPAN icon). This will open the parent window displaying the main menu (Fig. 4.1). As with all windows, you can reduce or expand this window (click and drag one of the corners) and move it about on the desktop (click on the title bar at the top and drag). There are 5 items on the main menu, whose functions are described below. The

menus are context-sensitive and will change depending on what type of window is currently active (as the result of menu choices which open new windows or switching among existing open windows). Menu choices can be made by clicking or double clicking with the mouse or by hitting the underlined letter key shown on the menu. The types of windows available will be explained below.

4.1.1 Main menu choices

The main menu choices are available when no window is active. When you click on a menu item, a submenu (drop-down menu) appears. As you drag down the sub-menu list, the submenu items are highlighted and a brief explanation of the item appears in the status box at the bottom of the window (Fig.

4.1). The main menu choices and their sub-menu items are as follows:

- File** This has 3 submenu items (Fig. 4.1): **New** lets you open a new empty Editing window for composing a command (.POP) file. When you select this item you get a standard Windows file dialog box so you can select a name and directory for the new file. If you select an existing file you are warned that it exists because the new file will wipe it out when you save. **Open** lets you create a new Editing window and load an existing .POP file into it. It gives you a file navigation dialog just like New. **Exit** quits RUNPOPAN and returns you to the Windows Program Manager. Exit will not work until you close all open windows (it keeps prompting you to close windows until all are closed).
- Popan** This has two sub-items: Clicking on **Run** when no Editing window is active will open a dialogue box displaying a list of all the .POP files in the default directory. You can choose one of these to run, or change directory to locate the .POP file you wish to run. Once you have selected a file, you click OK (or hit the Enter key) to run the task. A standard I/O window will open and display the progress of the task, paragraph by paragraph, just as you saw when you ran POPAN in batch. When the job finishes AND YOU HAVE CLOSED THE I/O WINDOW, the interface presents you with an empty RESULTS window (it will have a title such as RESULTS OF TESTANAL.POP) and the menu items will have changed. The menu items when a RESULTS window is active are explained below. The second sub-item is **Load**. Choosing this gives a file dialog box to choose a .POP file and after doing so, a split window is activated showing the corresponding .LOG and .RES file (either pane may be blank if a file is missing; an error warning occurs if both are missing). If there is no error, the active window is a RESULTS window and the menus change (see section 4.2.2 below).
- Other** This has the single sub-item: **Run MRint**. This opens a window to run the MR windows interface to the (DOS) programs JOLLY and CAPTURE. These programs are described briefly below and more fully in Section 5.
- Wizards** At the moment, this has only a single sub-item: **Entry**. This opens a window for running the ENTRY program which is used to enter descriptions of your sample times and data format and your raw data. It can generate data in forms acceptable to POPAN and to the JOLLY and CAPTURE programs. Additional notes on using ENTRY are given at the end of this section. Additional wizards (e.g. for selecting analysis models) are planned for later releases of POPAN
- Window** This is always on the menu bar, no matter what type of window is active. This lets you arrange the currently open windows in standard ways (Tile, Cascade) and lists all the currently open windows by title (type and file name). It displays a check beside the currently active window. You can switch the active window at any time by choosing from the Window list (drag and release over the highlighted window name).

The MR interface was written by Aaron McGlinchy and Richard Barker and is a Windows program that lets users specify a data input file and program options for either the open population analyses of program JOLLY (Pollock et al. 1990) or the closed model analyses of program CAPTURE (Rexstat et al. 1992). Both are DOS (character based) programs but choosing Other...RUN MR Interface launches a Windows interface that constructs the required input file from the data and options you specify. It then runs the DOS program and lets you browse the results. It makes the programs easier to use and extends their capabilities to some extent: for example it permits use of the common Release format for the data. The ENTRY wizard can be asked to produce data in this format.

ENTRY is a stand-alone Windows program that can be launched from within RUNPOPAN (using Wizards...ENTRY) or from outside (select its icon from Start...Programs...Popan-5 or double click its icon in the POPAN-5 folder). It is the first of several Wizards planned for POPAN: it leads you step by step through the process of specifying your sample meta-data, providing BACK and NEXT buttons so that you can correct errors or specify different data. It constructs an appropriate form for you to enter data and it checks the data you enter for correctness. When you are done it constructs a POPAN CREATE task that you can run (from RUNPOPAN) to create the binary file that POPAN needs to analyse your data. The CREATE

task can be edited later to add to or modify the data.

4.2 Running an existing POP file from the interface.

In this tour we will open an Edit window with one of the Test Suite command files, browse the POP file and look at Help for the paragraphs used, and then run it from the interface, and browse the results. In so doing, you will see how the menus change when an Edit window, a Help window, or a Results window is active. The tour consists of the following steps:

(a) Open an Edit window with an existing POP file

From the main menu (no windows open), first maximise the window (the middle of the 3 buttons in the upper right corner of Fig. 4.1; for Win 3.1, it is an up arrow) so it takes over your screen. We recommend that you ALWAYS maximise the main menu window to give you the largest possible working area but that you NOT maximise any sub-windows opened later. The reason is that a maximised window within a maximised window is ugly and confusing; for example there will be two Window title bars with their control buttons, one below the other, and it is easy to close the application by mistake rather than the sub-window. The sub-windows can be resized and arranged and are easiest to keep track of if they retain their (user-managed) size and position.

From the maximised main window, choose **File...Open**. Navigate to the \POPAN5\TEST directory (you may have to change drives as well as directories) and scroll through the list of .POP files. Select TESTANAL.POP by clicking on it to highlight it and then click on the OK box. The dialog box goes away and an Edit window opens with the title C:\POPAN5\TEST\TESTANAL.POP (an Edit Window is always titled by the full path of the POP file currently loaded). If necessary, re-position and widen the window until you can see a full (80 character) line, but it is not necessary to maximise the window. Now note that the menus have changed. Browse through the menu choices as follows: select (click on) each of the menu items in turn, and run down its list of sub-items. Don't release the mouse (or you will select the submenu item) but just highlight the items in turn and note the status bar short explanation for each. Before releasing the mouse run back up to the menu bar. Before resuming the tour, we explain these menu items:

4.2.1 Edit Window menu choices

- | | |
|--------------|--|
| File | This now has 4 submenu items. Open lets you open a another Editing window (you can have multiple Edit windows open at once). If you want another new (empty) window, use Open but type in a new file name at the file dialog box rather than choosing an existing POP file. Save lets you save the current window contents to disk (overwriting the file whose name is in the Title Bar) , and Save As lets you write the contents to another, new or existing, file whose name and path are specified from a file dialog box. Close closes the window (prompting you to save if the file has changed) and activates the next window in the windows list or , if this was the only window open, returns you to the main menu (Fig. 4.1). |
| Edit | This has the usual editing functions, Cut , Paste , Copy . The Edit window is a full editing window that lets you modify the text of the POP file by clicking to set the insertion point, dragging to highlight text, typing, deleting, etc., and by using these menu items to move or copy highlighted text to another position in the same or a different Edit window. See the next section (4.3) for examples of editing text. The last submenu item is Templates . Dragging down to this causes a fly-out sub-submenu to appear, listing the available templates (one for each paragraph type). Examples of Template use are given in section 4.3. |
| Popan | This has 3 items that work similarly to the Popan menu item on the main menu. Run causes POPAN to be launched to process the current contents of the POP file. If the Edit window contents has changed, you will be prompted to save the changes before the run proceeds. Load opens a new Results window and loads the results of the last run of this POP file into it (if the corresponding LOG and RES files exist); this window then becomes the active one, although |

your current Edit window remains open. **Load Another** gives you a file dialog box to choose any other POP file for which you want a new Result window opened. Both Load functions are useful if you want to see the results of a previous run of this or another POP file before resuming editing.

Window is the list of currently open windows, as described for the main menu (section 4.1.1).

Help gives a list of Help files available. There is one for each paragraph type as well as some additional topics (Raw History formats used with Create; Analysis Numbers used with Analysis and Simulate, etc.). Choosing Help opens a Help window, as described in the next section.

(b) Browse the .POP file in the Edit window.

Return to the tour by making sure the C:\POPAN5\TEST\TESTANAL.POP Edit window is active (choose it from the Window list if you accidentally opened any other window while browsing the menus). Scroll down through the window. Read the comment lines that begin with a C in the first position. Note that each paragraph begins with a paragraph name that ends with a colon: (e.g. SELECT:) and that this is then followed by several statements of the form Keyword = Keyword_Value ; and that the paragraph is terminated by a / right after the last Keyword statement. Other rules for POPAN task syntax are given below (section 4.3). In this case, the first paragraph is a SELECT to point to an existing binary file (BURN24.BIN) in the Test Suite, followed by two ANALYSIS paragraphs to fit two different models to these data. To get a fuller understanding of what these paragraphs do, we will open and browse the ANALYSIS Help file.

(c) Use the Help files.

Click on the Help menu and drag down until Analysis is highlighted. Release the mouse and a window with the title Help Analysis opens. You may have to reposition and widen the window to see a full line. Helpfile lines are less than 80 characters wide. Read through the Help file. It begins with a paragraph explaining the scope and purpose of Analysis. This is followed by a Task Definition section that begins by explaining any special features of the keywords. The section numbers (e.g. 5.1.3) refer to sections in the full manual (AS&B) from which this material is extracted. The task definition section then gives a detailed description of each keyword (in the order they appear on the Template). Scroll through this section, reading a few of them, until you come to the Examples section. The examples listed in Help are the same as those in the Test Suite. Now we will look at what the Help window menus do:

4.2.2 Help Window menu choices

File This has 2 submenu items. **Copy** lets you copy any currently highlighted text to the clipboard for later pasting into an Edit Window. A Help Window is read-only (you can't make changes to it), but Copy lets you extract parts of the Window for use in building a command file. For example, you could copy a paragraph from one of the Examples at the end of a Help file, or you could copy a Keyword=Keyword_Value line from the Task Definition section (although these would usually need some editing after the paste to select the desired Keyword_Value). **Close** closes the Help window and re-activates the Edit window from which Help was called.

Window is the list of currently open windows, as described for the main menu (section 4.1.1). You can have several Help windows open at once by returning to the Edit window and choosing Help again. You can also switch back to the Edit window by clicking on any portion of the window which is visible.

(d) Run the commands in the current Edit Window and browse the Results.

Now close the Help window and return to the Edit Window. If you have been following the tour exactly, this should be the only window open (check the **Window** menu) and its title should be

C:\POPAN5\TEST\TESTANAL.POP. If it isn't, close all windows until you are back to the main menu; then repeat the **File...Open** from step (a) to re-open the Editing window.

With the C:\POPAN5\TEST\TESTANAL.POP Edit window active, choose **Popan...Run**. A standard window will open (takes over the entire screen) and you will see the progress of the run, just as you did in batch mode (Fig. 3.1). When you see the END OF PROCESSING message, close the parent (Watcom windows application) window. The active window is now a Results window that RUNPOPAN has just opened with the title RESULT OF C:\POPAN5\TEST\TESTANAL.POP. The window is empty, and you need to load the results into the window by using **Popan...Load**. The results are loaded into a split window with the LOG file in the top pane and the RES file in the bottom pane. Initially, you will probably only see the source identification for both (the date / time stamp and name of the POP file used). The split bar between the 2 panes can be moved up or down; if you move it all the way up the LOG file pane is closed and you get a single window with the RES only. To restore the split bar, you have to close the window and reLOAD. Similarly, by dragging the split bar down, you can close the lower RES pane. For now, adjust the split bar to make the panes about equal in height and resize or maximise the window so you can see at least the full width (80 characters) of the LOG file in the upper pane.

First, scroll through the upper .LOG pane. As before (section 3.1.2), it contains the author / title section, the Run limits section, and then the parsed task commands. After this, you will again find the END OF RUN STATISTICS including a SUMMARY OF DIAGNOSTICS GENERATED and a SUMMARY OF PARAGRAPHS ENCOUNTERED. In this case, no errors should have been generated and the program should have encountered and executed 1 SELECT and 2 ANALYSIS paragraphs.

Now turn to the lower RES pane and begin scrolling through its output. When you come to tables with lines longer than 80 characters you will have to use horizontal scrolling too. Scroll through the SELECT output (which is quite long in this case because there are 24 sample times and none of the options to suppress parts of the SELECT output was chosen) until you come to the first ANALYSIS paragraph. After STATISTICS table Definitions, look for the STATISTICS TABLE itself. In this experiment, note that samples were not large ($N(I)$ mostly around 50 or less) but that the marked fraction ($M(I)/N(I)$) and recovery rates ($R(I)/S(I)$) are quite substantial (generally over 60%). These rates have to be reasonably high if simple Jolly-Seber estimates are to have any chance of decent precision.

Scroll down until you are positioned so the ESTIMATE TABLE is at the top of your screen. You should see the column headings $AH(I)$, $MH(I)$, $NH(I)$, $S(NH(I)|N)$, etc. The table is quite wide and you will have to use horizontal scrolling to see all of it. The estimates for abundance ($NH(I)$) are fairly good, but those for survival ($PHI(I)$) are often inadmissible and those for births are rather imprecise (the standard errors $S(BH(I))$ are as big as the estimates $BH(I)$). This is fairly typical for the Jolly-Seber model when applied to sample times that are quite close together. POPAN has methods of resolving these problems (using $GROUP=$ in SELECT and $POOLING=$ in ANALYSIS) which you can try later (see the command files for TESTSELS and TESTANPO). The Jolly-Seber model is discussed in Section 5.2.1 and 5.2.2 of AS&B.

Continue scrolling through the output for the second analysis. This is an even more general model than the previous one (it allows for some forms of age-dependence and type-1 loss effects on survival; it is discussed in Section 5.2.4 of AS&B). As a later test program shows (see command file TESTSTAT), there is no evidence of such effects occurring in these data, so the estimates are actually somewhat more imprecise than with the first analysis.

You usually won't need to use the menus when viewing a Results window but for completeness, we describe their menus.

4.2.3 Results Window menu choices

File This has 3 submenu items. **New** and **Open** let you activate a new (empty) Edit window or and Edit window with an existing POP file chosen by file dialog just as in the Main menu. **Close** closes the current split window. Usually, this returns you to the Edit window from which the Run was launched. In general it returns you to the window from which the original Result Window **Load** was done, which may be the Main menu.

Popan This has sub-items **Run**, **Load**, and **Load Another** which are like their equivalents on the Main or Edit window menus. **Run** (re-) runs the most recently run POP file. **Load** lets you re-load the contents of the current results window and **Load Another** opens a new window (useful for comparing this run with a previous one).

Window Lets you see the list of currently open windows and switch to another window.

When you are finished browsing through the .RES file, close the window. At this point, the C:\POPAN5\TEST\TESTANAL.POP Edit window should be active. We will finish this part of the tour by making a new POP file by altering an existing one and saving it to a new file before running it. This is a quick way to try variations of an analysis without creating a command file from scratch.

(e) Edit, Save or Save As, Run

Start by creating a new sub-directory to take the altered file. You can't do this from within RUNPOPAN so iconise the maximised window: the leftmost of the 3 control buttons (the down arrow next to the double arrow in Win 3.1) in the upper right corner of the main window (Fig. 4.1). The window closes and the POPAN icon should appear on your Taskbar (at the bottom of your desktop in Win 3.1). Use My Computer or Windows Explorer (the File Manager program in Win 3.1) to make a new folder (directory) under C:\POPAN5 called MYTOUR. Close or iconise the File Manager and click (double click in 3.1) the POPAN icon to re-activate the program.

We might want to try a constant survival model since the survival rates were all very high. This might improve precision and fix the inadmissible estimates. To do this, we will re-use the SELECT and the first ANALYSIS, with alterations. Start by getting rid of the second ANALYSIS task. Click at the beginning of the comment that begins C THE SECOND ANALYSIS USES... and drag to the end of the file. The second ANALYSIS paragraph is highlighted. Hit the backspace key to delete it. Now the last line reads:

```
DILUTION = PRESENT ; LOSSES = PRESENT ; POOLING = NO ;/
```

Edit it so that it reads:

```
DILUTION = PRESENT ; LO = FI ; POOLING = NO ;/
```

If you want to see the choices available for these keywords, consult **Help...Analysis** or **Help...Analysis Numbers** (note that keywords and pre-defined keyword values can be truncated to 2 or more letters).

While we are at it, let's suppress most of the SELECT and ANALYSIS output, since we've already seen it. Add the line

```
OUT=TABLE ; SUPP=BOTH ;
```

after the TITLE keyword line (don't add it after the end of paragraph mark /) and in the SELECT paragraph above it, change WRITE = BOTH to WRITE = NONE. Finally, change JOLLY-SEBER FULL in the comments

and titles to JOLLY-DICKSON CONSTANT PHI (or just delete these now inaccurate phrases). Your edited file should then look something like this:

```
SELECT:
```

```
TITLE = 'SELECTING ALL ANIMALS AND ALL SAMPLE TIMES' ;  
INPUT = 'TEST\BURN24.BIN';  
WRITE = NONE ;/
```

```
C THE FIRST ANALYSIS USES THE JOLLY-DICKSON CONSTANT PHI MODEL
```

```
ANALYSIS:
```

```
TITLE = 'TEST ANALYSIS...' ;  
OUT=TABLE ; SUPP=BOTH ;  
DILUTION = PRESENT ; LO = FI ; POOLING = NO ;/
```

Make sure the punctuation is correct: every statement ends in a ; and each paragraph ends in ;/ and

single quote delimiters are around the TITLE and INPUT values. Now choose **File...Save As** to navigate to your new directory (C:\POPAN5\MYTOUR and save to a new file called MYTESTAN.POP. **NOTE:** if you don't want to clobber your original file (in this case, C:\POPAN5\TEST\TESTANAL.POP) it is important to do **Save As** before you **Run**. If the file has changed, RUNPOPAN will prompt you if you want to save before running, but then if you respond Yes, it will do the Save and replace your original file. Now do the Run (**Popan...Run**), and close the standard output window at the END OF PROCESSING. The analysis can take a while on slower machines because it involves an iterative fit (it took about 30 seconds on a 33 Mhz 386 with no co-processor). Open a Results window (**Popan...Load**) and browse the RES file.

Check the LOG file in the upper pane to see that both the SELECT and ANALYSIS worked. If not, read the error messages and switch to the editing window to fix them. If the run worked, scroll to the end of the lower pane and look at the Estimate Table. For Jolly Dickson analyses, there is a second part to the table to add additional estimates (here, the capture rates, PH(I)). Note that this analysis did get rid of the inadmissible PHI values. Note too, that the analysis takes account of the unequal sample spacing. PHI estimates are equal only if the inter-sample times are equal (see the T(I) in the STATISTICS TABLE). For longer intervals, the survival rates are lower, as they should be. This model did not, however, resolve the poor precision and inadmissible values in the Birth estimates. Perhaps the model is wrong (survival rates aren't really equal, or no births are occurring) or it is not possible to estimate the very small mortality and birth values that occur in short sample time intervals (see TESTANPO for further ways to analyse this data set). You might want to use **Popan...Load Another** to compare these results to those obtained before (from TESTANAL.POP).

4.3 Building a command file from scratch

In this section, we will show how to build a POPAN command file from scratch in the editing window with the help of the templates. The template for a given paragraph type lists each keyword and saves you the trouble of typing the Keyword= part or the punctuation. You simply fill in or select the Keyword_Values you want. The template provides drop down choice lists when the possible Keyword_Values are predefined (e.g. in the ANALYSIS paragraph, LOSSES = PRESENT or ABSENT or HETEROGENEOUS) or gives you a text box to fill in where they are not (e.g. DATASET = text;). It provides a short help phrase for each Keyword (in the form of a Comment) and inserts punctuation (at the end of statements and paragraphs) and delimiters (e.g. the quotes and / or parentheses required for some Keyword_Values). This reduces the chance of making syntactic errors, but full checking that each paragraph is correct is not done; the parser may still find errors when the command file is RUN. When a template is completed, the user can Paste (with or without closing the template) and a complete paragraph is inserted in the active Edit window at the current insertion point. A command file can be built up by defining some first paragraph (typically using the CREATE or SELECT template) and Pasting it, then opening another template to analyse the data (e.g. ANALYSIS, STATISTICS, UFIT) and in turn using Paste to append it to the file. At any point, when not in a Template window, one can edit the contents of the Edit window, including the deletion, modification, copying, or moving of text including that pasted from a template. As in the previous section, when you are satisfied with the command file, you Save the file. Then, as before, the file can be run and its results browsed. If there are errors, the file can be re-edited and Run again.

You can, of course, create any command file from scratch using a text editor like Notepad outside RUNPOPAN, and save it to the appropriate POPAN sub-directory for the relevant 'project'. In either case, it is the user's responsibility to make sure that the command file is syntactically correct and that the data input files required by the command file at RUN time (.RAW and .BIN data files) are present in the specified 'project' subdirectory. Therefore, before proceeding with the tour, we outline the main syntax and file specification rules for POPAN. A full discussion is in AS&B (section 1.4)

4.3.1 Syntax of command statement statements and file specification

- A paragraph begins with a paragraph name (ending in a colon), and is followed by one or more keyword statements (ending in a semicolon). The paragraph ends with a / which must follow the last ;

- with no space between. The last statement may be null (;/).
- Paragraph names and statements are only scanned for in columns 2 through 72. The first column is for the C of comment statements, and characters beyond column 72 are ignored. (the Template helps you see where column 72 is). You can now change this to longer lines using the LENGTH keyword. There can be multiple statements per line, and single statements can span multiple lines. Blank lines and non-significant blanks can be used freely for readability.
- Upper or lowercase characters may be used, but the scanner translates all characters to uppercase before parsing, and POPAN translates all raw data attribute characters to upper case; in particular, this means that you cannot use upper and lower case of the same letter as distinct attributes.
- Keywords and pre-defined keyword values can be truncated down to 2 (or more) letters.
- When a keyword value requires real values, there must be at least one digit before and at least one digit after the decimal point (e.g. 1.0 rather than 1.)
- Keywords INPUT and DATASET are used consistently in all paragraphs to specify the input and output file, respectively. The Keyword_Value for these Keywords should be a relative or absolute path \file.ext enclosed in single quotes (supplied by the template). A relative path is relative to the file in which the POPAN.EXE executable resides (NOT relative to where the .POP file resides). Thus, for example INPUT= 'SKUNKS\JUNE.RAW' will try to open the file C:\POPAN5\SKUNKS\JUNE.RAW.
- Users should adhere to the conventions on file extensions given above (section 2.2). LOG and RES files must be kept in the same directory as the POP file that created them if they are to be found by the RUNPOPAN interface.

4.3.2 An example with use of templates.

(a) Start a new Edit window

Close all windows and start this tour from the main menu. Make sure the main menu window is maximised. Select **File...New** from the Main menu. When the file dialog box appears, navigate to your new subdirectory MYTOUR and enter NEWJOB.POP in the file name box. If this file already exists, you will be warned (because a later Save could obliterate it). An empty Edit Window titled C:\POPAN5\MYTOUR\NEWJOB.POP will be activated. We will create a program similar to TESTCR8V.BIN but it will analyse a somewhat smaller dataset to save typing. It will consist of a CREATE paragraph followed by a couple of ANALYSIS paragraphs. Start by typing some explanatory comments so that you know later what the POP file was for. Remember to type C in the first position of each comment line. Type something like the following:

```
C My First Example (NEWJOB.POP)
C This simple example reads the Rodli Tarn data (Ricker, 1975 p. 81)
C in-line using grouped CMR histories using minimal metadata.
C It then applies some 2-sample (closed model) ANALYSIS tasks to them.
```

(b) Make the first task paragraph using the CREATE Template.

Now, with your cursor on the line after your last comment, click on the **Edit** menu, then on **Template**. The list of paragraphs drops down. Click on **Create**. A new Template window opens with the title:

Create paragraph for C:\POPAN5\MYTOUR\NEWJOB.POP

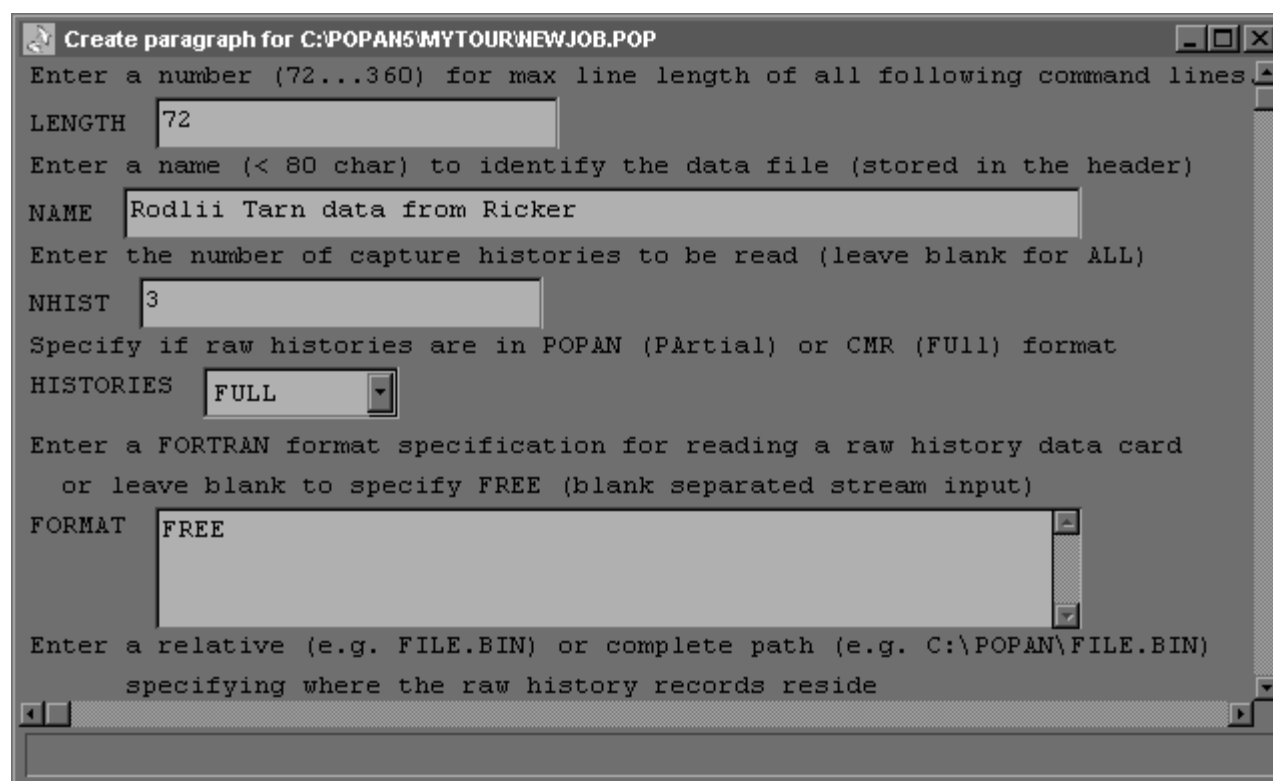
(since you can have multiple copies of the same template open, this identifies the Edit window it belongs to). Reposition and widen the Template window if necessary so you can see a full line. The template should look like that in Fig. 4.2, below, except that most of the entry boxes are empty. Templates have 3 types of entry boxes:

- 1• A Text box (like the NAME and NHIST boxes in Fig. 4.2) lets you enter a text string. It scrolls to the right for long strings. Arrow keys can be used to scroll back.
- 2• A Scrolling Box (like the FORMAT box in Fig. 4.2) lets you enter very long strings that will be broken up into multiple lines when pasting back to the Edit window. It has vertical scrolling bars and text

wraps within the box (you can also use the return key to force a new line at a given point...this can be useful for some Keyword_Values that have natural break points).

3. A choice box (like the HISTORIES box in Fig. 4.2). This will show the default choice when a keyword has a number of pre-determined keyword-value choices. Clicking in the box at the right side (Fig. 4.2) drops down the complete list of choices. A selection is made by dragging down to the desired entry and releasing when the entry is highlighted Or you can click on the arrow box on its right side to drop down the list and click on the desired selection. In either case the drop-down menu disappears and the current selection shows in the box.

Note that above each entry box is a comment that briefly explains how to fill in the box. Choose **File...Defaults** from the menu. It's a good idea to always select this before filling in a Template but only at the beginning unless you want to start again (choosing this resets all boxes to their defaults, losing any entries you have made). You will see that defaults are filled in for some of the boxes (ALL for NHIST, and FREE for FORMAT, etc.). Now fill in the various boxes:



Create paragraph for C:\POPAN5\MYTOUR\NEWJOB.POP

Enter a number (72...360) for max line length of all following command lines

LENGTH 72

Enter a name (< 80 char) to identify the data file (stored in the header)

NAME Rodlii Tarn data from Ricker

Enter the number of capture histories to be read (leave blank for ALL)

NHIST 3

Specify if raw histories are in POPAN (Partial) or CMR (Full) format

HISTORIES FULL

Enter a FORTRAN format specification for reading a raw history data card or leave blank to specify FREE (blank separated stream input)

FORMAT FREE

Enter a relative (e.g. FILE.BIN) or complete path (e.g. C:\POPAN\FILE.BIN) specifying where the raw history records reside

Figure 4.2: The top of the CREATE Template after filling in the fields for NEWJOB.POP

1. Click in the NAME box and type: Rodli Tarn data from Ricker
2. Replace the contents of the NHIST box by highlighting (drag over) the word ALL and type: 3
3. Choose FULL instead of PARTIAL in the HISTORIES choice box.
4. Scroll down until you can see the IDENTIFIER keyword and choose GROUPED instead of NUMBER.
5. In the boxes below, type 1 in the BEGIN box and 2 in the END box (these must BOTH be specified when you use CMR history format (HISTories=FULL)).
6. Since we aren't going to enter any of the meta-data, you might think you are ready to paste this paragraph into your edit window. Select **Paste Close** from the **File** menu. You will get a dialog box that says:

You have to enter something for ANUM

- 7• Scroll down to the ANUMBER box and enter: 0
- 8• Repeat the **Paste Close**. This time the template window closes and your edit window should look like this:

```

C My First Example (NEWJOB.POP)
C This simple example reads the Rodli Tarn data (Ricker, 1975 p. 81)
C in-line using grouped CMR histories using minimal metadata.
C It then applies some 2-sample (closed model) ANALYSIS tasks to them.
C*****BEGINING OF CREATE PARAGRAPH*****
CREATE:
  NAME           ='Rodli Tarn data from Ricker';
  NHIST          =3;
  HISTORIES      = FULL;
  FORMAT         =FREE;
  INPUT          =CARDS;
  IDENTIFIER     = GROUPED;
  BEGIN         =1;
  END            =2;
  ANUMBER        =0;
  DATASET        =TEMP ;
;/
C*****END OF CREATE PARAGRAPH*****

```

Note that the Create paragraph has been pasted in at your insertion point. It begins and ends with banner comments that also mark the maximum line length allowed (the last asterisk is in column 72). You may have to post edit the results of a paste to break lines at more appropriate or readable points than the paste does, and you can go back and add comments or additional statements. You can pull up the help window for this paragraph by using **Help...Create**. This CREATE is not complete because INPUT=CARDS; requires that in-line data be supplied immediately following the paragraph. Do this by placing the insertion point JUST AFTER the ;/ that terminates the paragraph (second last line above; you mustn't leave the insertion point at the end or POPAN will try to read the last comment card line as data) and hit return. Then enter the following 4 lines (they may begin in column 1):

```

52 1 0
120 0 1
57 1 1
$ENDDATA

```

The first line specifies that 52 animals were captured in sample 1 but not in sample 2; the second that 120 animals were captured in sample 2 but not in sample 1; the third that 57 animals were captured in both samples; the \$ENDDATA must come at the beginning of the last line to terminate the data. NOTE: check that the keywords HISTORIES, FORMAT, IDENTIFIER and ANUM are specified exactly as shown above; the form of the histories that POPAN expects depends on these 4 keyword values and if the keywords are not those shown above, the program can fail (see online Help...Raw Histories for details)

- (c) Add two ANALYSIS paragraphs to the end of the Edit window.

Now position your cursor after the last line and hit the Enter key to start a new line. Choose **Edit...Template** and select the **Analysis** Template. Most of the ANALYSIS keywords are choice boxes and all have defaults. This time Fill in a title (e.g. First Analysis using...). Now choose **File...Paste Close**. You should see that the following has been added to the end of your Edit window:

```

C*****BEGINNING OF ANALYSIS PARAGRAPH*****
ANALYSIS:
  TITLE          ='First analysis using...';
  LOSSES         =PRESENT;

```

```

DILUTION      =PRESENT      ;
;/
C*****END OF ANALYSIS PARAGRAPH*****

```

This is actually a valid ANALYSIS paragraph (the interface has added the two choice keywords to ensure a valid paragraph even if you specify nothing) but these defaults will select the Jolly-Seber Full model. We can't use this model with our data because it requires at least 3 sample times. Delete this text (drag over to highlight it and press backspace); to see the default values, open the Analysis template again. Notice that RUNPOPAN has remembered the TITLE you entered. This time, choose **Option...Include defaults** and then **File...Paste Close** again. This time the following should be added to your window:

```

C*****BEGINNING OF ANALYSIS PARAGRAPH*****
ANALYSIS:
  TITLE          ='First analysis using...';
  WRITE          = NONE;
  CEFFORT        = ABSENT;
  LOSSES         = PRESENT;
  DILUTION       = PRESENT;
  POOLING        = NONE;
  TRACE         = NO;
  OUTPUT         = BOTH;
  SUPPRESS       = NONE;
  SAVE          = NONE;
;/
C*****END OF ANALYSIS PARAGRAPH*****

```

This is still the Jolly-Seber Full model, with all choice options specified explicitly...note that the text defaults (keywords SIZE and ITERATIONS) are not included because we did not choose **File...Default** when we opened the Template. Delete this ANALYSIS paragraph as before. Now we will paste the 2 paragraphs we really want. Re-open the template and note that **Option...Include default** is no longer checked. Now, from the template, select LOSSES=ABsent and DILUTION=ABsent and choose **Paste** from the file menu. The template stays open so you can prepare another paragraph of the same type. Now, change the TITLE to replace "First" by "Second" and select CEFFORT=PResent. Choose **File...Paste Close**. The two paragraphs will now be at the end of your Edit window and should look like this:

```

C*****BEGINNING OF ANALYSIS PARAGRAPH*****
ANALYSIS:
  TITLE          ='First analysis using...';
  LOSSES         =ABSENT;
  DILUTION       =ABSENT      ;
;/
C*****END OF ANALYSIS PARAGRAPH*****
C*****BEGINNING OF ANALYSIS PARAGRAPH*****
ANALYSIS:
  TITLE          ='Second analysis using...';
  CEFFORT        =PRESENT      ;
  LOSSES         =ABSENT;
  DILUTION       =ABSENT      ;
;/
C*****END OF ANALYSIS PARAGRAPH*****

```

This completes our introduction to the use of Templates. You can now do a **File...Save** and **Popan...Run** (followed by a **Popan...Load**) to see the results of the program if you wish. Browse the results

as before; you should get a population estimate of 338.4 (S.E. 25.5) from the first analysis and an estimate of 358.9 (28.6) from the second. Note that the capture rate estimates (PH(I)) are the same for the second analysis since the model constrains them to be constant. If the program fails, check that it looks like this, and if not, edit it and Run it again:

```
C*****BEGINNING OF CREATE PARAGRAPH*****
CREATE:
  NAME           ='Rodli Tarn data from Ricker';
  NHIST          =3;
  HISTORIES      = FULL;
  FORMAT         =FREE;
  INPUT          =CARDS;
  IDENTIFIER     = GROUPED;
  BEGIN          =1;
  END            =2;
  ANUMBER        =0;
  DATASET        =TEMP ;
;/
52 1 0
120 0 1
57 1 1
$ENDDATA
C*****END OF CREATE PARAGRAPH*****
C*****BEGINNING OF ANALYSIS PARAGRAPH*****
ANALYSIS:
  TITLE          ='First analysis using...';
  LOSSES         = ABSENT;
  DILUTION       = ABSENT ;
;/
C*****END OF ANALYSIS PARAGRAPH*****
C*****BEGINNING OF ANALYSIS PARAGRAPH*****
ANALYSIS:
  TITLE          ='First analysis using...';
  CEFFORT        = PRESENT ;
  LOSSES         = ABSENT;
  DILUTION       = ABSENT ;
;/
C*****END OF ANALYSIS PARAGRAPH*****
```

We conclude this section with a description of the menus when a Template window is active and some general rules on how Templates operate.

4.3.3 Description of Template menus and operations.

- | | |
|---------------|--|
| File | This has 5 submenu items. Paste copies the current template contents into the Edit window but leaves the Template active (can be changed and pasted again). Paste Close copies the current contents and closes the Template window, returning you to the Edit window. Clear returns the template to its original state (all fields are cleared and defaults showing in choice boxes only). Default clears all fields and resets them to the default values, if any. Close closes the Template window and re-activates the Edit window from which the Template was called. |
| Option | This has two choices: Include comment causes the comments above each entry box to be included in any subsequent paste. Include default causes all fields for which there are explicit defaults to be included in any subsequent paste. Both of these are toggle values that can be selected or deselected at any time (a check appears by the menu item if it is currently on). The values are NOT remembered if the Template is closed and re-opened. |

- Window** Lets you see the list of currently open windows and switch to another window.
- Help** Lets you open a Help window for the current Template (there will only be one item on the Help menu).

The following notes point out some less obvious features of templates:

- 1• Before opening a template, make sure your cursor is positioned OUTSIDE any existing paragraph. Paste is done to the current cursor position, so you can get a paragraph (incorrectly) within another paragraph if you are not careful. If you do make this mistake, use cut and paste to move the just pasted paragraph to its correct position before proceeding.
- 2• Most text boxes do NOT require the inclusion of the delimiters (parentheses and /or single quotes) as these are pre- and post-fixed when the contents are pasted. If delimiters are required (usually because there is more than one form for the Keyword_Value), the comments will make this clear.
- 3• With scrolling boxes, the line breaks on pasting do not correspond to where the text wraps in the template (except when you use the Enter key to force line breaks). Always check the pasted text for all long Keyword_Values and edit them to put the line breaks in more suitable spots. This is especially important for file specifications and format strings. Be careful not to let continuation lines put text in the first (comment) column or beyond column 72 (unless you have increased the line LENGTH).
- 4• If you try to paste or close a template before filling in obligatory keywords, you will be warned (one keyword at a time). We saw an example of this above with the CREATE keyword ANUMBER. This only catches unconditionally required keywords such as ANUM; some keywords are conditionally obligatory (a keyword that must be specified if another keyword takes a particular value: BEGIN and END in CREATE are examples of this) and it is up to the user to satisfy these requirements, although POPAN will catch these at run time and generate an error.
- 5• When you paste a Template, the entry values are recorded. If you close the Template and re-open it later, the recorded values will be restored (i.e. will show in the entry boxes). If you had never pasted, the Template will be clear when it is re-opened, even if you had changed the values before closing it.
- 6• If you try to close an Edit window that has open Templates, you will get a warning and the close will be aborted. Note that the title bar for a Template window (and hence its title listed under the Window menu) gives the Edit window it is attached to.
- 7• In a new or cleared template, the choice fields will show the default values but these will not be pasted unless you have enabled **Edit...Include defaults**. Defaults for text fields are not shown unless you choose **File...Defaults** when the template is opened. We recommend you always do this when you open a new Template. Defaults that are variably generated by POPAN (e.g. LIST: FORMAT=) depending on .BIN header values or other keywords are not shown (just leave these boxes blank to get the default).
- 8• If you want the pasted values to include the defaults and /or comments (shown on the template) you must choose **Edit...Include defaults** and /or **Edit...Include comments** before pasting and *you must choose them again* after you re-open the template.
- 9• You may have multiple templates open for a given Edit window (including multiple instances of the same template) and multiple instances of a template belonging to each of several Edit windows. Just remember that the Window menu can be consulted to see what is open and that the Template title includes the name of the Edit window that will receive the Paste from that template.

5. Other programs and wizards callable from RUNPOPAN

5.1 The ENTRY program.

The ENTRY program aids in the input of raw mark-recapture data for the POPAN program. It is an interactive Windows program that uses the same windowing interface as RUNPOPAN. It takes you step by step through the process of defining a CREATE paragraph in POPAN and then prompts for the capture history raw input data in the format appropriate to the type of identifier, number of attributes, and sample times and history format that you have specified. Like a (Microsoft) wizard, it provides BACK and NEXT buttons to let you go back and forth to view or change previously specified values. At the end of a successful interactive session, it generates a (.POP) file with the CREATE paragraph (including the data in-line, as with the Rodli Tarn example in section 4) and a save file (.ESV) file that lets you resume using the wizard with your saved values at a later date. You cannot actually Run the CREATE task from within ENTRY. You have to do a batch RUN (as shown in section 3.1), or a Popan...Run from within RUNPOPAN, specifying this .POP file. The run will then CREATE the binary (.BIN) file from these files. This use of ENTRY completely eliminates the need to write a POPAN CREATE paragraph or an external raw data file. It also ensures that fairly comprehensive attribute and capture history checks are done as the data are entered, thus ensuring (usually) that the CREATE task will run without errors.

The Entry SAVE file can also be useful to users who maintain their data in computerised form using a spreadsheet or database program. You can use ENTRY to specify the sample descriptions and the first few histories. Then the .ESV file can be opened with a text editor (e.g. Notepad) and it should be clear what form additional histories must take. The histories can be exported to a text file from the spreadsheet in appropriate format (most spreadsheets can save to a text file using a space delimited /formatted text format). You can then copy the histories from the text file and paste them into the .ESV file. The ESV file is then reloaded into ENTRY and the attribute and sample times will be checked as it is loaded. Because the ESV is read in free format, only the order of history fields matters, not their spacing.

ENTRY will also produce a data file in Release format. This is the form needed by the Release program of Burnham et al (1987) but is also the form accepted by program Mark (White, 1998; encounter history form LLLL) and by programs JOLLY (Pollock et al. 1990) and CAPTURE (Otis et al. 1978, updated by Rexstad et al. 1992). CAPTURE provides comprehensive models for closed populations, including models allowing for heterogeneity that are not possible with open population models. CAPTURE and JOLLY can be used from within POPAN through the MR interface under the Other menu (see section 5.2).

ENTRY allows the capture history to be specified in the two different forms (POPAN form, or CMR form as explained below) and as individual histories or grouped histories. Histories can have 0 to 8 single-character attribute codes (see AS&B Section 2.1.4 or Help: Raw Histories). Individual histories have a (numeric or character string) identifier; this is replaced by a group count for grouped histories, indicating how many animals share this history (and set of attributes).

POPAN also permits you to designate that an animal is Lost on Capture (LOC), meaning that it is known NOT to be in the population after the last capture time; it also permits designation of an animal as injected (INJ) when it is known NOT to be in the population before its first capture time. Program JOLLY can account for LOC but not INJ, and program CAPTURE allows neither INJ nor LOC. ENTRY provides check boxes to specify either LOC or INJ for any given history.

5.1.1 Background on POPAN history formats

Although ENTRY sets up the appropriate data entry form for histories, you need to know a bit about the different history formats that POPAN can accept. In the POPAN data form, the capture history is indicated by the number of captures (T) and a variable length list of capture times (X(i), i=1,...,T). In the CMR data form, the capture history is given as a fixed length list (D(i), i=1,...,L where L is the number of sample times determined by the CREATE keywords BEGIN and END as $L = \text{END} - \text{BEGIN} + 1$). The i-th

element of this list is 1 if the animal was captured at time i , and 0 if not captured. For example, an animal that is seen at times 1, 6, and 7 in a 10-sample experiment would have:

T=3 X = 1 6 7 (POPAN format)
 D = 1 0 0 0 0 1 1 0 0 0 (CMR format)

Note that the D list can be typed without spaces for a FORMATTed read but ENTRY uses an unformatted (FORMAT=FREE) read so elements of the history must be blank-separated (by one or more blanks). Similarly, attributes in POPAN must be encoded by a single character, and for a FREE read, must be enclosed in single quotes. If the first attribute designates sex (e.g. M for males, F for females, U for unknown) and the second age (e.g. 1 for juveniles, 2 for young, 3 for adult), then the complete history for an individual might be:

123 'M' '1' 3 1 6 7 in POPAN format,
 123 'M' '1' 1 0 0 0 0 1 1 0 0 0 in CMR Format.

(animal with numeric identifier 123 is a male juvenile captured 3 times at times 1, 6 and 7. If you have specified ID=CHAR, the ID must also be enclosed in quotes. If you have specified ID=GRouped, then the identifier is the group count and must be a whole number; the above history would then indicate that 123 male juveniles had this capture history. The ID=GRouped method is more convenient when batch marks are used and / or very large samples are taken. In these situations, it is not unusual to have a sample of hundreds or thousands of animals (most of which are never seen again) and it would be tedious to enter these as individual histories. On the other hand, grouped histories don't permit recording of individual identifiers such as tag number; this can make it more difficult to collate and update the data as each new sample is taken.

The ENTRY program can be launched as a stand-alone application or from within RUNPOPAN. Any of the usual ways will work to launch ENTRY on its own: double click on its icon from My Computer, create and use a shortcut; or select **Start... Programs... Popan-5... ENTRY** or select **Start...Run** and type C:\POPAN5\ENTRY at the Open prompt. Launching from the RUNPOPAN interface is done by selecting **Entry...Run**. You will also need to decide where you want to save the (.POP, .ESV) files created by ENTRY. By default, it will save to your POPAN5 directory (where entry.exe resides) but you can use absolute or relative path names to direct to a more appropriate folder. Before you start the program make sure you know where you plan to place these files and that files with the name you plan to use do not already exist in that directory (ENTRY will over-write files).

5.1.2 An Introductory tour of ENTRY

ENTRY has been organised into two phases: meta-data specification and data (history) entry. Start ENTRY from the RUNPOPAN main menu by choosing Entry...Run. When ENTRY starts up, you are presented with the first dialog box (Fig. 5.1, below).

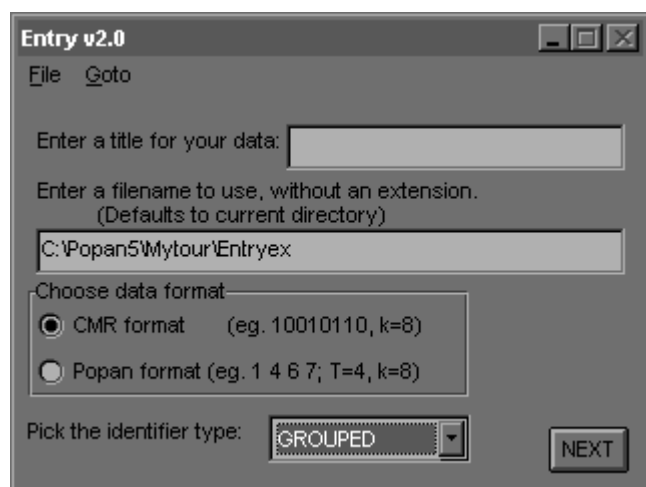


Figure 5.1: The first phase ENTRY screen

This lets you specify a title that will be stored in the binary file (keyword NAME in CREATE), and select the data format (fixed length CMR format, or variable length POPAN format) and identifier type. The filename can be a fully specified path (as shown in Fig 5.1) or just a file name. This will give the name and location of both the .POP file (here, Entryex.pop) and the .ESV file (Entryex.esv) which will be stored in the same folder (the Mytour subfolder of the Popan5 directory). You **can't** give a relative path (e.g. Mytour \Entryex) but you can give a non-existent folder (you will be asked if you want to create the new folder).

Fill in the dialog box so it looks like Fig 5.1 (modify the path appropriately for your POPAN directory) and then click on NEXT.

The next dialog box asks if you want to specify absolute sample times. With CMR histories you should answer YES. When asked for the number of sample times, type 4 and click the OK button. The next dialog box lets you specify the absolute time (day number) of each sample. It's important to specify these so that POPAN can adjust birth and survival rates to account for unequal sample time spacings. Give midpoint times if samples are spread out over time. The default times, 1.000, 2.000... are already filled in for you as shown in Fig. 5.2 and you can use these if the sample times are unknown to you.

Figure 5.2: ENTRY's sample time specification dialog

dialog box you type a single character code followed by the code label enclosed in single quotes. Type m'male' and hit Enter. Then enter the remaining 2 codes as f'female' and 'unknown' (you must type a single space before 'unknown': this is a valid POPAN code. The attribute and its codes then appears in the text box, as shown in Fig. 5.2. Note that all attribute codes are translated to UPPERCASE. Now click ADD Attribute again and add the attribute Grid and its 2 codes as shown in Fig. 5.3. If you make a mistake, click DELETE Attribute, select from the list of attribute names the one you want to delete; this deletes the attribute and all its codes. You can't selectively correct attribute codes and labels, but you can re-enter them. You can experiment with adding and deleting attributes, but when you have them specified as shown in Fig. 5.3, hit the NEXT button.

Figure 5.3: ENTRY's Attribute specification dialog.

You can add a time to the list by typing it in the Enter box and clicking ADD TIME. This will insert the new time (if different from the existing times: times must be unique) and increase the sample time count to 5. To remove a time, click to highlight it in the Sample Times box and click REMOVE TIME. Try adding a time like 2.5 as shown and removing it again. Then move on to attribute specification by hitting the NEXT key.

You will be asked if you have any attributes to specify. If you say NO the next dialog box is skipped. So click YES. You will see a dialog box like Fig. 5.3 but with the text area empty. Click ADD Attribute and when prompted for a name, type SEX in the text box. When prompted for the number of codes, type 3 (sic!). In the next

You have now specified all the meta-data and ENTRY is ready to create a data entry form. ENTRY brings up a dialog that first asks if you want to allow for injections and losses on capture on this form. In this case click in "Allow for loss on capture" but not in "allow for injections" (you can click again to toggle your choice). If you have chosen the GROUPED identifier type (see Fig. 5.1) this is where you can also specify that you want a Release format output file. You can still specify this choice if you have losses on captures, but not if you have injections. Check the box that specifies Release format histories because we will use these later (in section 5.2).

When you click NEXT, ENTRY warns you that this is your last chance to check and revise

your meta-data before generating the data entry form. For practice, click NO and go BACK through the previous dialog boxes and come back again by clicking through the NEXT buttons.

#	Group Count	Attribute 1	Attribute 2	Attribute 3	Attribute 4	Attribute 5	Attribute 6
# 1	36	'M'	'3'	1	1	1	1
# 2	14	'M'	'3'	1	1	0	1
# 3	27	'M'	'3'	1	1	1	0
# 4	34	'M'	'3'	1	1	0	0
# 5	9	'M'	'3'	1	0	1	1
# 6	22	'M'	'3'	1	0	0	1
# 7	19	'M'	'3'	1	0	1	0
# 8	99	'M'	'3'	1	0	0	0
# 9	11	'M'	'3'	0	0	1	1
#10	38	'M'	'3'	0	0	0	1
#11	35	'M'	'3'	0	0	1	0
#12	15	'M'	'3'	0	1	1	1
#13	14	'M'	'3'	0	1	0	1
#14	28	'M'	'3'	0	1	1	0

ID (group count)	Attributes	Histories (CMR Fmt k=4) eg: 1001
60	1: M 2: 3 Lost: ☐	0100

Figure 5.4: ENTRY's data entry form for the Entryex example. The form is at the bottom; entered data (with sequence numbers) is shown in the read-only scrollable region above.

This is a good time to SAVE your (Entryex.esv) file. Use File...Save to save. You will be warned if you are about to over-write a file. If you don't want to over-write, you can say NO to the prompt and choose a new file name by going BACK to the first dialog box (see Fig 5.1). If you File..Quit at this point, you can return to this point with all information restored by restarting ENTRY and immediately using File...Load in the first dialog box (Fig. 5.1) after launching ENTRY. If you have started entering data, the Load will take you to the data entry screen: you can no longer edit the meta-data except by editing the .esv file itself (see below).

The data entry screen (Fig 5.4) has customised entry boxes at the bottom for entering a new history in the form you have specified by your meta-data. Histories can be entered from the keyboard without using the mouse: use the <Enter> key to leave a text field and <Tab> to skip over the Injected / Lost on capture check boxes.

Use the space bar to toggle the check boxes. After you enter a fixed length field such as an attribute code, the cursor jumps to the next field automatically. You can however, use the mouse to position to any field at any time.

Hitting <Enter> when the cursor is in the last (Histories) field causes the History to be checked and written to the read-only display area above. To delete a history, use the Delete menu. A dialog box will let you specify the sequence number (#) of the history you want to delete and then remove it from the data. To change a history you have to delete and re-enter it.

Enter the 15 histories shown in Fig. 5.4: don't forget to enter the last history shown in the entry form (just before hitting the <Enter> key). Check all 15 histories against Fig. 5.4 and then use **Finished...Create paragraph** to create the final Entryex.pop file. When prompted don't quit and return to RUNPOPAN because we also want to save the .REL file. Use **Finished...Make a .rel file** to do this. Then use **File..Quit** to leave Entry. You will be prompted to do a final save of Entryex.esv. This will bring you back to the main RUNPOPAN menu.

From RUNPOPAN, use **File...Open** to open Entryex.pop in the MYTOUR folder. The file should look like this:

```
C Create paragraph made with Entry v2.0
C Using entry save file C:\POPAN5\MYTOUR\ENTRYEX.esv
CREATE:
  LENGTH = 320 ;
  NAME = 'Simulated data for Entry Example' ;
```

```

NHIST = 15 ;
FORMAT = FREE ;
INPUT = CARDS ;
HISTORIES = FULL ;
BEGIN = 1 ; END = 4 ;
IDENTIFIER = GROUPED ;
SVALUE = (1.000000,2.000000,3.000000,4.000000) ;
TCHECK = BOTH ;
ANUMBER = 2(3,2) ;
ACHECK = YES ;
ALIST = SEX,Grid ;
AVALUES = SEX (M) 'male' (F) 'female' ( ) 'unknown'
          Grid (0) 'Grid0000' (3) 'Grid0037' ;
DATASET = 'C:\POPAN5\MYTOUR\ENTRYEX.bin' ;
SAVE = ASIS ;/
36 'M' '3' 1 1 1 1
14 'M' '3' 1 1 0 1
27 'M' '3' 1 1 1 0
34 'M' '3' 1 1 0 0
9 'M' '3' 1 0 1 1
22 'M' '3' 1 0 0 1
19 'M' '3' 1 0 1 0
99 'M' '3' 1 0 0 0
11 'M' '3' 0 0 1 1
38 'M' '3' 0 0 0 1
35 'M' '3' 0 0 1 0
15 'M' '3' 0 1 1 1
14 'M' '3' 0 1 0 1
28 'M' '3' 0 1 1 0
60 'M' '3' 0 1 0 0

```

\$ENDDATA

You can run this as is, which won't produce very interesting output but will create the binary file needed for later runs. To make this more interesting, add an ANALYSIS to get a preliminary look at the estimates; you don't need a SELECT because POPAN assumes you want to SELECT the binary just CREATED: Type the following 2 lines at the end of the file (on the line after \$ENDDATA), SAVE it, and do a Popan..Run:

ANALYSIS:

TITLE = 'Jolly-Seber model ignoring (meaningless) attributes';/

After this has run and you have dismissed the Watcom Window, you can use POPAN...Load to browse the LOG and RES file from this run. This concludes the tour of ENTRY. If you want to do further analyses of these data you don't need to re-run the CREATE paragraph; you SELECT the binary file. For example here is a POP file that lists the binary file in raw history format:

SELECT:

TITLE = 'Selecting complete file.. ';

INPUT = 'C:\POPAN5\MYTOUR\ENTRYEX.bin' ;/

LIST:

TITLE = 'listing raw histories, using default format';

C Grouped histories can be ordered by Cell Number (see Help..List)

TYPE = ORDER ;

ATTRIBUTES = YES;

OUTPUT = LISTING; /

5.2 The DOS stand-alone programs.

We distribute 2 additional programs with POPAN5: JOLLY for open population analysis, and CAPTURE for closed population analyses. Both are character-based DOS programs that require an input file that specifies program options and gives the data (inline). When run using this input, the program produces an output listing of the results. The MR Interface that can be launched from the Other menu of RUNPOPAN makes it much more convenient to run these 2 programs because it:

- lets you specify the options using a point and click dialog box, then generates the appropriate input file for you
- lets you specify an input data file in release (.rel) format and checks for correct history formats. The release format is somewhat more general than that required by the programs, so the interface massages the data into the correct form. This means that one input file can be used with either program.
- generates an input file (a .jol file for JOLLY, a .cap file for CAPTURE), combining the input data and program option statements, runs it, and lets you browse the resulting output file (.out). The input file is saved for later re-running stand-alone (see below for methods). The output file is also saved so that it can be browsed or printed later.

We give a brief description of each program and describe how it behaves when run stand-alone. We then give a tour of using the interface.

5.2.1 Program JOLLY

The JOLLY program provides a number of standard analyses and model tests using grouped (CMR format) mark-recapture data. The analyses and tests are bundled together so that the user only has to specify a model choice and a few things about the raw data to obtain a quick first analysis. These analyses include goodness of fit tests, confidence intervals, and average parameter values that are more difficult to get from POPAN alone. There is a comprehensive manual (Pollock et al 1990) that explains these features and gives much valuable advice on the planning of mark-recapture experiments and on the analysis of the data and interpretation of the analyses and tests. The form of input file is explained in Section 9 of their manual but most of the keywords and form of the raw data are illustrated in the examples (ENTRYEX.JOL and VOLES.JOL) which is distributed with the current release and should have been copied to the JOLLY subdirectory of your Popan5 folder (e.g. C:\POPAN5\JOLLY\ENTRYEX.JOL).

To run the JOLLY program standalone, use the DOS prompt, locate to the JOLLY subdirectory of your POPAN5 folder and type JOLLY. You can also start it from START...PROGRAMS...POPAN-5 (you should be able to see the JOLLY icon) or from the START...RUN box (type C:\POPAN5\JOLLY\JOLLY.EXE in the Open box). The program requires no command line arguments. The opening screen presents a list of 7 command prompts that looks like this:

```
Program JOLLY <01/27/91>
```

```
-----
```

- 1) Data filename:JOLLYINP
- 2) Name of print file:LPT1
- 3) Models:A,B,D
- 4) Print definitions:NO
- 5) Print Var-Cov of N:NO
- 6) Run...
- 7) Quit

```
Choice(1...7)?_
```

As an example of its use, here is how you would run the VOLES example to produce the results given in Pollock et al (1990), Tables 4.7 and 4.8 :

Step 1. Type 1 at the Choice? prompt and hit return. You will get a new prompt line:

Enter new filename: _

at which you enter:

voles.jol

On hitting return, JOLLY will reflect this choice at the 1) prompt and return the cursor to the Choice? prompt. Note that the JOLLY program is resident in your POPAN5 folder and you can use relative paths (as above) or absolute path names (e.g. c:\popan5\jolly\voles.cmr).

Step 2. Choose 2 at the Choice? prompt, and when prompted, fill in:

voles.out

Step 3. Choose 6 at the Choice? prompt and hit return. The program will then execute and direct your output to the file specified in Step 2

The model choices (Prompt 3 above), and their POPAN equivalents are given below. See AS&B, Table 5.1.1 or Help: Analysis Numbers for choice of ANALYSIS models and Table 8.1.5 of AS&B or Help: Test Numbers for TEST choices.

JOLLY	POPAN	ANALYSIS:				TEST:				
Model	Description	NU=	CE=	DI=	LO=	#	DI=	LO=	TD=	CE=
A	Jolly-Seber Full (Birth and Death)	1	AB	PR	PR	4	PR	PR	AB	AB
A'	Jolly-Seber Death-only	5	AB	AB	PR	3	AB	PR	AB	AB
B	Jolly-Dickson Full, fixed P (Cap. prob.)	9	PR	PR	PR	12	PR	PR	AB	PR
D	Jolly-Dickson Full, fixed P & PHI (survival)	12	PR	PR	FI	-	(no equivalent)			
2	Jolly-Seber Full, modified for heterogeneous survival	3	AB	PR	HE	-	(no equivalent)			

Take a look at VOLES.JOL or ENTRYEX.JOL to see the form of an input file for program JOLLY. Note that both start with a number of control lines and then include the data inline following the FORMAT statement. JOLLY expects each data line to start with the fixed length capture history in CMR format followed by the group count; FORMAT allows these elements to be picked out of a line that is not ordered this way or which contains extraneous elements (like attributes, which JOLLY does not allow for). The MR interface will construct a complete .JOL file for you out of a release (.REL) format history file. JOLLY allows for losses on capture, but not for injections.

5.2.2 Program CAPTURE

POPAN provides only limited analyses for closed populations (no birth or death), and very limited analyses with allowance for capture heterogeneity. There are many closed model analyses, and because these models do not allow for losses and new entries, it has been possible to develop abundance and capture rate estimates that are robust to various forms of heterogeneity. Given this, a particularly successful sample design is the robust design (see Pollock et al, 1990). The design is composed of closely spaced secondary samples within each of several widely spaced primary samples. For example, one could sample the first 5 days of a month and then leave off sampling for the rest of the month, and repeat this for several months. Closed models are used to estimate abundance over the secondary samples within each primary period, and open models are used to estimate survival and new entries between primary periods (months).

Thus there is a natural fit between the open models of POPAN and the CAPTURE program that provides the most comprehensive suite of closed model analyses (Otis et al. 1978, updated by Rexstad et al. 1991) .

To run CAPTURE stand-alone, you have to prepare an input file with a combination of control and data lines. The data must not include either losses-on-capture or injections (unlike the closed models in POPAN), and CAPTURE doesn't allow for group counts (all histories are for one animal). CAPTURE is then run from the Start...Run menu of Windows or from the DOS prompt, using command-line arguments to specify the input and output files (which should be in the same folder as the program). For example:

```
capture i=taxi.cap o=taxi.out
```

The control lines describe the data (number of samples, etc.) and analysis options including several different estimation models, tests for closure and model selection. It is much easier to specify these things using the MR interface and let it generate the control lines. It will also take a Release format (.REL) input file and use it to construct the inline data, propagating out histories with group counts >1 and filtering out losses-on-capture. The taxi data in the example above are described in the Otis et al. (1978) paper and are distributed with POPAN; you can look at taxi.cap in the POPAN5\CAPTURE folder to see a typical input file.

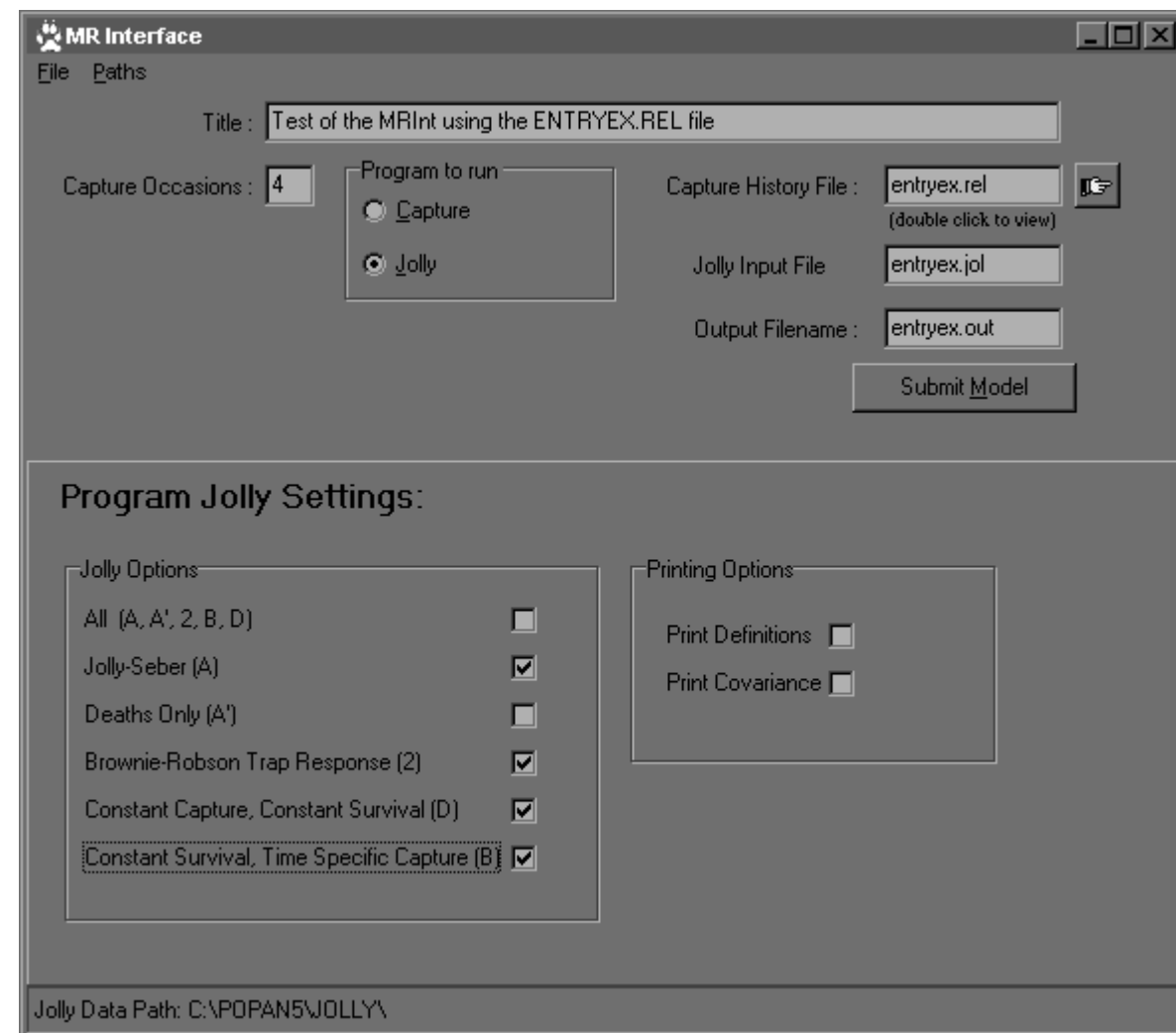


Figure 5.5: The MR Interface after completing specification of data and program options for the JOLLY program.

5.2.3 The MR Interface for running programs CAPTURE and JOLLY

Launch the interface from RUNPOPAN by choosing Other...Run MR Interface. You can also run it from outside POPAN using START...PROGRAMS...POPAN-5...Other. The interface requires that you specify the number of sample occasions and an input Capture History File and choose either JOLLY or CAPTURE. Figure 5.2 shows the interface for the JOLLY program. After typing in the name of the File, the program fills in the names for the Input and Output file. After choosing the program, the lower panel lets you select the program options available for that program:

Before filling out the panel as shown in the figure, you may have to set the paths. The installation will set these correctly for a standard install (to C:\POPAN5) but if you installed to a different drive or directory, you need to select the correct paths. Choosing the Path menu gives you a dialog box (not shown here) that lets you browse the folder hierarchy and set the correct the paths for programs and (input/output) files. MRint assumes a common folder for both JOLLY and CAPTURE input files (Capture History Files) which by default is set to the Other subdirectory of your POPAN5 folder (e.g. C:\POPAN5\Other). You will probably want to change this for most runs. It expects to find the programs in the JOLLY and CAPTURE folders respectively and you should only have to set this once. The input and output files for JOLLY (.jol and .out) go to the JOLLY folder by default, and the input and output for CAPTURE (.cap and .out) go to the CAPTURE folder. Note that Capture History Files contain only data lines in Release format, whereas .jol and .cap input files are generated by the Interface and combine data and control lines.

To run the ENTRYEX example using JOLLY do the following: Launch MRint, and choose the Path menu. Click in the Capture Histories Path: box and navigate the file hierarchy box to the right to go up to POPAN5 (double click) and then down to the Mytour folder (double click again). Check that the other 4 paths to the programs and location of input/output files are correctly specified as described above, and change them if necessary. Click on the Save and Exit button to return to the main panel and fill it in as shown in Fig. 5.5 (be sure to fill in both the upper and lower panel). Click on the Submit Model button.

A DOS window opens up and you see the dialog described in section 5.2.1 flash by (MRint has created the correct responses for you). When it comes to a halt, dismiss the DOS window. You should see a dialog box that asks you if you want to browse the output.

When you have run this successfully, you can try the CAPTURE program. There is an example file called taxi.inp in the Other folder. From the main panel, click the little hand to the right of the Capture Histories File. This gives a dialog box that lets you browse for the input file. If you select **Files of Type: All files** from the drop-down menu at the bottom of the dialog box, and navigate to the Other folder, you should find a file called Taxi.inp. Select this, and then fill in 10 for the number of capture occasions. You still have to make sure that the path to the Capture History file is set correctly, so use the Path menu and set the Capture History path (first path box) to the Other folder under POPAN5. Return to the main dialog panel (click SAVE and EXIT) and choose some model options from the lower panel. Then submit the model.

Again a DOS box is launched to run the program. When you see the Normal termination message, you can dismiss the DOS box and browse the output file.

NOTES:

1. There is a bug in MRint when choosing the Chao model analyses. The program analysis statement generated by MRint for these models has a syntax error that causes CAPTURE to ignore the analysis. If you want these analyses, just select the All (models) option. The correct task statement is:

Task Population Estimate Mt-Chao

but the interface generates Chao M(t) instead. You can correct these statements in your .cap file for running again later. Look for updates (see McGlinchy, 1998, for web site) that correct this problem

2. The MRint interface will not work under Windows 3.1; you must have Windows 95 or Windows NT 4 or later versions of Windows. The rest of RUNPOPAN (including the Entry Wizard) continue to work under Win 3.1/32S and you can run JOLLY and CAPTURE standalone under DOS as describes in section 5.2.1 and 5.2.2.

6. ODDS AND ENDING

6.1 Notes on use.

The following is a more or less random collection of suggestions for using POPAN effectively.

- Keep your command files short. This has the advantage that the program will run fairly quickly, that .LOG and .RES files will be reasonably short and so, can be browsed through or printed more quickly. Typically, a command file should consist of a CREATE alone (getting the errors out of your raw data is enough work for one job), or of a SELECT (of a previously created .BIN file) with a couple of analyses (ANALYSIS, STATISTICS, TEST or UFIT tasks).
- Get to know the Test Suite POP programs and use them to apply to your own data. Often you can use them as is just by changing the input file in the SELECT. You can copy a paragraph from one of the examples and paste it into the POP file you are building; this is especially useful for the contingency table tests constructed with ANALYSIS: Number=39 (see TESTAN39.POP and TESTTEST.POP). You can do the same with the detailed constraints printed to the RES file when you use global constraints in UFIT (see section 7.1 of AS&B).
- Use SELECT: with SAVE= sparingly. This is intended for *permanent* saves of a subset or sample transformation of the SELECTed INPUT file. Normally, such a subset is only of interest for the duration of the subsequent ANALYSIS or STATISTICS tasks. POPAN will quickly generate this 'on the fly' and delete the temporary file when the job completes if you use SAVE=NO (the default). Saving the dataset permanently is slow, especially if it has to be sorted (SAVE=Ordered), and the .BIN files can quickly fill up your disk.
- In the current release, there is no reason to sort the data files, except to order them by identifier for LISTing purposes. Sorting was required for the data maintenance operations in POPAN-2 and POPAN-3 (MODIFY and ADD) but as MODIFY is not implemented in POPAN5 and ADD is rarely used, there is no need for it here. Moreover, the SORT can be very slow. LIST can produce the sorted .RAW data from an unsorted .BIN file.
- If you find the restriction on sample times at ANALYSIS time constraining, remember that when the file is CREATED it can actually have up to 512 sample times. You must reduce this to the limits given in each LOG file only for ANALYSIS, STATISTICS, and LIST which you can do using BEGIN, END, GROUP, and OMIT in SELECT. This can be done repeatedly to analyse the data in blocks of sample times with little loss of information. See AS&B, Section 8.1.2.7, for an example of this (using TEST, but ANALYSIS could be substituted in the example given there).

6.2 Errors and what to do about them.

POPAN catches syntactic/semantic errors in the .POP file when it is parsing and writes the errors to the .LOG file. Execution time errors are written to the .RES file (if POPAN catches them in time). If you obtain errors that you cannot fix, we would like to know about them, but we would appreciate receiving a brief written description of the problem, along with a diskette or (emailed) file containing the .POP program that produced the error along with any input files required by the job. We cannot, however, promise to fix all errors reported, but will do our best to fix them by the next release.

1. Input file not found. Input .BIN and .RAW files must exist in the directory specified on the INPUT= keyword. If they don't, you will get an error message (severe) that includes the name of the file being searched for. If you are using unit numbers, the program aborts immediately.
2. Fatal I/O errors when trying to read raw data in CREATE. The order, number and type (integer, character, real) of variables on each data line depends on the choice of keywords (particularly IDENT, FORMAT and HISTORIES) and any errors can cause unrecoverable problems. Using the ENTRY

Wizard should eliminate these problems completely.

3. Multiple accesses to a file. A file can be used many times in a single POPAN job (.POP file), but there are some restrictions on multiple access. A binary file can be read many times (by SELECT...see TESTLIST.POP for an example) but it must not then be used for output (over-written). A single .OUT file can be used repeatedly to store table output. .OUT files are not re-wound, and tables are just added to the end of the file but they can't be used for input. .TBL files, used as the output DATASET by ANALYSIS or SIMULATE, and used as INPUT to UFIT are closed after each use so they may be re-used in a later task (see TESTMVAL.POP); if re-used for output, the previous value is over-written.
4. Unrecoverable execution time errors. POPAN attempts to check wherever possible to ensure that zero divides, logs or square roots of negative numbers, etc., do not occur. If they do, however, the program may terminate catastrophically. The places where this is most likely to happen is when the Jolly-Dickson constrained-parameter models (see Table 5.1.1 in AS&B or Help: Analysis Numbers) or UFIT are applied to inadequate or inappropriate data; typically, where capture probabilities are low, or turnover rates are high, or assumptions are seriously violated. In these cases, convergence of the iterative models is slow, or fails completely, or gives negative variances and other absurd estimates. Sometimes stratifying the data, eliminating a few particularly weak or troublesome sample times, or use of GROUP= in SELECT to reduce the number of samples, can help resolve the problem. In UFIT, it is worth trying the model several times with different starting points as problems are often in the early stages of the iterative search and the problems will go away later in the search.
5. Index of an array out of range and some execution time errors like those in 3. are the result of program errors, particularly uninitialised work variables. If you get errors like these that go away when the unchanged task is re-run, let us know about it (including the traceback information printed when the program fails).

References

- Arnason, A. N., and C. J. Schwarz. 1995. POPAN-5: Enhancements to a system for the analysis of mark-recapture data from open populations. *Applied Statistics*, **22**: 785-799 (Proceedings of the EURING94 meetings at Patuxent, MD).
- Arnason, A. N., and C. J. Schwarz. 1998. Using POPAN-5 to analyze banding data. (To appear) *Bird Study* (Proceedings of the EURING 97 meetings in Norwich, UK).
- Arnason, A. N., C. J. Schwarz, and G. Boyer 1998 *POPAN-5: A data maintenance and analysis system for mark-recapture data*. Technical Report, Department of Computer Science, University of Manitoba viii+318 p. (available from the anonymous ftp site: <ftp://ftp.cs.umanitoba.ca/pub/popan> or from the web site: <http://www.cs.umanitoba.ca/~popan>
- Burnham, K.P., G. C. White, C. Brownie, and K.H. Pollock. 1987. Design and analysis methods for fish survival experiments based on release-recapture. *American Fisheries Society Monograph* 5, 437 pp.
- Cooch, E. G., R. Pradel, and N. Nur 1997. *A Practical Guide to Mark-Recapture Analysis using SURGE - 2nd Edition*. Centre d'Ecologie Fonctionnelle et Evolutive - CNRS, Montpellier, France. 134 p. For this, and links to other population software sites visit the web site at <http://www.biol.sfu.ca/cmr/index.html>
- Lebreton, J.-D., K. P. Burnham, J. Clobert. and D. R. Anderson 1992. Modelling survival and testing biological hypotheses using marked animals. A unified approach with case studies. *Ecological Monographs* **62**: 67-118.
- McGlinchy, A. and R. Barker. 1998. M-R-I. Mark-Recapture Interface for programs CAPTURE and JOLLY. Landcare Research, Maths and Stats Department PO Box 69, Lincoln, Canterbury, New Zealand. For software updates go to <http://www.landcare.cri.nz/>

- Otis, D.L., K. P. Burnham, G. C. White, and D. R. Anderson. 1978. Statistical inference from capture data on closed animal populations. *Wildlife Monographs*, 62. 135pp.
- Pollock, K.H., J. D. Nichols, C. Brownie, and J.E. Hines. 1990. Statistical inference for capture-recapture data. *Wildlife Monographs* 107. 97pp. (In 1991, the monograph was available for \$US 4.45 from The Executive Director of the Wildlife Society, 5410 Grosvenor Lane, Bethesda, MD 20814; For software and manual updates for programs JOLLY and CAPTURE see the U.S. Fish and Wildlife Service (Patuxent) web site at <http://www.mbr.nbs.gov/software.html>)
- Rexstad, E. and Burnham, K. (1992). User's guide for interactive program CAPTURE. Fort Collins: Colorado Co-operative Fish and Wildlife Research Unit, Colorado State University. 29 pp.
- Ricker, W. E. 1975. Computation and interpretation of biological statistics of fish populations. *Bull. Fish. Res. Board. Can.* 191: 382p.
- Schwarz, C. J., and A.N. Arnason. 1996. A general methodology for the analysis of open-model capture recapture experiments. *Biometrics* 52: 860-873.
- Seber, G. A. F. 1982. The estimation of animal abundance and related parameters. 2nd ed. Griffin, London.
- Smart, J. 1996 WxWindows 1.66 A free C++ framework for development of multi-platform graphical applications from the same body of C++ code. Available from: <http://web.ukonline.co.uk/julian.smart/wxwin/>
- White, G. C. 1998 Program MARK. Available from: <http://www.cnr.colostate.edu/~gwhite/mark/mark.htm>